Modeling Complex Adaptive Systems and Complexity for Interactive Art

Christa Sommerer

Ph.D. Thesis

June 2002

Abstract

Complex System Sciences, as a field of research, has emerged in the past decade. It studies how parts of a system give rise to the collective behaviours of the system and how the system interacts with its environment. It approaches the question of how life on earth could have appeared by searching for inherent structures in living systems and trying to define common patterns within these structures. Complex Systems are also often described as systems where the whole is more complex than the mere sum of its parts, and these systems are also considered to be at the point of maximum computational ability, maximum fitness and maximum evolvability.

Several scientific models have simulated Complex Adaptive Systems. These try to model the emergence of complexity within computer-simulated environments inhabited by artificially evolving organisms. My objective in this thesis is to study the application of Complex Systems and Complex Adaptive Systems to Interactive Art and to test how one could construct interactive systems that can create dynamic and open-ended image structures that increase in complexity as users interact with them. Ideally, these interactive artworks should become comparable to Complex Adaptive Systems or even become Complex Systems themselves by satisfying some of the key properties of such systems.

Contents

5 - Artistic Interpretations of Artificial Life and Complex Adaptive Systems

6 - On-line Artworks Creating Other Forms of Complexity

7 - Modeling Complex Adaptive Systems and Complexity for Interactive Art

8 - Life Spacies II: Complex Adaptive Systems for Interactive Art

9 - Experiments in Modeling and Generating Complexity for Interactive Art on the Internet

1 - Introduction

1.1 - Background

The origin of this project lies in the fundamental questions of how complexity arises in living systems and how one could construct an interactive artwork that can model and simulate the emergence of complexity. Based on the insight that interaction and communication between entities of a system are the driving forces behind the emergence of higher-ordered structures, the purpose of my thesis is to apply principles of Complex Systems and Complex Adaptive Systems to the creation of interactive, computer-generated and audience-participatory artworks and to test whether complexity can arise within these systems.

Complex Systems are often described as systems where the whole is more complex than the mere sum of its parts, and these systems are also considered to be at the point of maximum computational ability, maximum fitness and maximum evolvability. They are situated between order and chaos, a stage often referred to as a "phase transition" or as "life at the edge of chaos". These systems generally display high entropy and high connectivity and are considered irreducible. It also appears that internal changes within these systems can be described by a power law distribution or a change of entropy, sometimes referred to as an "entropic force-flux pair", which regulates the flow of energy in time. Complex Adaptive Systems (CAS), on the other hand, are defined as systems that can self-organise, metabolise, self-reproduce, couple to each other, react to their neighbours and to external control, learn and adapt, expand their diversity, explore their options, and organize a hierarchy of higher-order structures.

While several scientific models have displayed features of Complexity or Complex Adaptive Systems in worlds inhabited by artificially evolving creatures, these systems have mainly remained closed to outside interaction parameters. The aim of my thesis is therefore to apply principles of Complex Systems and CAS to interactive artworks and test whether user interaction parameters can help to increase the systems' internal complexity and help to create artworks that are not pre-defined or pre-programmed but instead capable of creating a large and open-ended variety of image outputs that would meet the requirements of a complex system.

1.2 - Motivation

Modelling the emergence of complexity remains a major challenge in Complex Systems Sciences. Complex System Sciences is in fact not a fully established field, and definitions and properties are hence not completely defined or finalised. Complex Adaptive Systems, on the other hand, tackle this problem by modelling systems of artificially evolving agents that can metabolise, reproduce, self-organise, adapt and evolve, and create higher-ordered structures.

My thesis will therefore investigate the issue of Complexity through a two-fold path. First, I will test the possibility of applying principles of Complex Systems to interactive artworks by utilizing the techniques and principles of CAS and Artificial Life. My aim here is to create dynamic image structures that are not merely designed by the artist but can also develop, adapt and evolve through user interactions, to finally create higher-ordered structures that expand in diversity and explore their options. Ideally, these artworks should become comparable to Complex Adaptive Systems by satisfying some of their key properties. To achieve this goal, I will study the definitions, properties and principles of Complex Systems, their connection to the Origin of Life Theories, and their background in Artificial Life. Through this study, I will identify the key techniques to apply in creating the software and hardware structure of these works.

After analysing the results obtained through these works, I will explore other forms of complexity and test whether the Internet could be used as a large, dynamic and complex database with which users could interact to create complex systems that satisfy some of the other current complexity definitions. To do this I will program software structures in C, C++ and Java and design interfaces that allow multi-modal and intuitive access to image and sound data from the Internet. It is anticipated that as users interact with these systems, the systems' internal parameters should ideally inter-connect and self-organise, creating complex image and sound structures and hyper-links that are not pre-defined but instead unrepeatable and irreducible. Complex Systems are often described as systems that are more than the mere sum of their parts, situated somewhere between order and chaos. With these experimental systems I aim to test whether one can design an interactive and complex systems for the Internet that satisfies some of the outlined complexity measures and principles.

1.3 - Outline of Thesis

In Chapter 2, I will introduce the field of Complex System Sciences by briefly explaining its origin and defining what is currently understood to be a Complex System. I will then formulate some of the commonly used definitions of complexity and continue to describe the various properties associated with Complex Systems. I will then describe some of the major complexity features, such as Phase Transition, Self-Organised Criticality, Entropy, Emergence and the notion of Life at the Edge of Chaos and its link to Complex Adaptive Systems.

In Chapter 3, I will give a brief overview of some of the existing Origin of Life Theories and discuss the possible increase in complexity within the evolution of life. I will also draw a connection to computer simulations that try to model this increase in complexity by simulating emergence and self-organisation. This will lead us to Chapter 4, which examines Complex Adaptive Systems, a field in Complex System Sciences that is strongly rooted in Artificial Life. I will briefly explain the philosophy and background of Artificial Life and then introduce Genetic Algorithms and Genetic Programming, the main techniques used to create Artificial Life and CAS. In Section 4.3 I will describe several scientific simulations of CAS and analyse how the above principles are applied.

Chapter 5 will introduce some of the existing artistic interpretations of Artificial Life and CAS in art, generative music, generative design, generative architecture, and games. In Chapter 6 I will look at other forms of complexity in on-line art and see whether some of the selected on-line artworks display properties of complex systems.

In Chapter 7, I will finally outline my own principles and objectives for designing interactive artworks that apply principles of complexity for interactive art. In Chapter 8, I will describe my first interactive artwork called "Life Spacies II" and its earlier version "Life Spacies". For these systems, I applied the methods of creating CAS outlined in Chapter 4. I will give an in-depth description of these systems and their internal algorithms and parameters and show how the system's internal interactions are intertwined with the users' external interactions. I will show how a Complex Adaptive System for Interactive Art was created; this system models artificial evolution and increases its overall complexity through user interaction and satisfies most of the definitions of Complex Systems defined in Chapter 2.

Chapter 9 will describe on-line interactive artworks for which I do not directly model Complex Adaptive Systems but instead aim to model complexity in general by using the Internet as a platform. The first work I introduce in Section 9.2 is called "VERBARIUM". It is an interactive website where users can write text that is subsequently encoded into 3-D shapes. As more and more users interact with this system, these shapes become more and more varied, building up a collective image that increases in complexity as users interact with it. I will describe the text-to-form encoding algorithm in detail and also analyse the kind of complexity properties that emerge within this system.

The second system in this category of on-line works is called "Riding The Net". This interactive artwork, described in Section 9.3, also uses the Internet, but this time the Internet functions as a large and dynamic database of image and sound data that users can retrieve through their speech and touch interaction. By speaking into microphones and touching the images on an interactive screen, users can re-organise, re-link and prioritise this information, which becomes an instance of the users' current preferences and their interactions with this large and dynamic database. While only some features of Complex Systems emerge within this system, it will function as a testing ground for how to model increasingly inter-connected interactive systems for the Internet in the future. Two follow up projects, called "The Living Room" and "The Living Web", allow users to physically immerse themselves in dynamic and complex information spaces to create, re-organise and inter-connect complex image and sound data through their own interactions. These systems will be described briefly in Section 9.4.

In Chapter 10, I will then analyse all of the results obtained in Chapters 8 and 9 in light of the initial purpose of my thesis. These findings will be evaluated by comparison with the properties of the artworks and systems outlined in Chapters 4, 5 and 6. I will finally point to future research directions and outline possible applications of the results obtained.

2 - Complex System Sciences

2.1 - What is Complex System Sciences?

Complex System Sciences, as a field of research, has emerged in the past decade. It approaches the question of how life on earth could have appeared by searching for inherent structures in living systems and trying to define common patterns within these structures. A whole branch of research—not only within biology but also across its borders to physics and computer science—deals with complex dynamical systems and can be seen as the attempt to find basic organizing principles. Related efforts have been made to define the notions of complexity and organization quantitatively (Aschby 1962; Baas 1994; Bennett 1988; Cariani 1992; Chaitin 1992; Jantsch 1980; Kauffman 1993; Landauer 1988; Langton 1989; Pagels 1988; Wicken 1987; Wolfram 1984; Yates 1989).

Complex System Sciences studies how parts of a system give rise to the collective behaviours of the system and how the system interacts with its environment. Social systems forming, in part, out of people, the brain forming out of neurons, molecules forming out of atoms, and the weather forming out of air currents are all examples of complex systems. The field of Complex Systems Sciences cuts across all traditional disciplines of science as well as engineering, management, and medicine. It focuses on certain questions about parts, wholes and relationships. These questions are relevant to all traditional fields. There are three interrelated approaches to the modern study of complex systems: (1) how interactions give rise to patterns of behaviour, (2) understanding how to describe complex systems, and (3) the formation process of complex systems through pattern formation and evolution (Yaneer, 2000).

2.2 – What are Complex Systems?

Although there is no exact definition of what a Complex System is, there is now an understanding that when a set of evolving autonomous particles or agents interact, the resulting global system displays emergent collective properties, evolution, and critical behaviour that exhibits universal characteristics. Such a system is fundamentally novel and not deducible into its mere parts. These agents or particles may be complex molecules, cells, living organisms, animal groups, human societies, industrial firms, competing technologies, etc. All of them are aggregates of matter, energy, and information that display the following characteristics. They:

- couple to each other
- learn, adapt and organize
- mutate and evolve
- expand their diversity
- react to their neighbours and to external control
- explore their options
- replicate
- organize a hierarchy of higher-order structures

2.3 - Existing Definitions of Complexity

To find a common principle behind the organisational forces in natural systems is a complex task, and it seems as if there are as many theories as there are theorists. Some of the numerous theories on Complex System shall be briefly mentioned here. Valuable information on the various approaches and definitions are taken from Edmonds (1999).

2.3.1 - Algorithmic Information Complexity - The KCS Definition

The best known definition of complexity is the KCS (Kolmogorov-Chaitin-Solomonoff) definition (Kolmogorov, 1965), (Solomonoff, 1964), (Chaitin, 1966) describing Algorithmic Information Complexity (AIC). It places complexity somewhere between order and randomness; that is, complexity increases as Pmin (the shortest algorithm that can generate a digit sequence, S) increases to a length equal to the sequence to be computed. When the algorithm reaches this incompressibility limit the sequence is defined as random. The main properties of the Algorithmic Information Complexity as described by Chaitin (1966) are:

1. The more ordered the string, the shorter and hence less complex the program.
2. Incompressible strings (those whose programs are not shorter than themselves) are indistinguishable from random strings.
3. Most long strings are incompressible.
4. In a range of formal systems one cannot prove (within that system) that there are strings above a certain fixed level of complexity (derived basically from the AIC of its axioms).
5. In general it is incomputable.

The KCS definition thus distinguishes between "highly ordered" and "highly complex" or "highly disordered" structures. Especially property 2 in Chaitin's list shows the deep connection between AIC and disorder or randomness.

2.3.2 - Computational Complexity

Computational complexity is now a much studied area with many formal results (Von Neumann, 1956), (Papadimitriou, 1994), (Fagin, 1973). The foundation of complexity theory is the research into computability theory undertaken from the 1930's onward by Alan Turing, Alonzo Church and Stephen Kleene, among others. The primary considerations then were the formalization of the notion of a computer (e.g., the Turing machine, Church's lambda calculus) and whether such computers could solve any mathematical problem.

Computational complexity is then considered the asymptotic difficulty of computing a solution for a problem class once a program has been found for a solution (it is measured usually either by the time it takes or by the memory that is needed). In Computational complexity the "difficulty" of the solution is therefore strongly linked to the problem size.

2.3.3 - Arithmetic Complexity

Arithmetic Complexity concerns itself with finding the minimum number of arithmetic operations needed to perform a computational task and is thus often applied to optimization problems in engineering. Nadenau and Reichel (2000), for example, describe an arithmetic complexity measure for an image compression algorithm that can optimize the number of arithmetic operations, memory demands, and bandwidth. Arithmetic Complexity is more a practical definition than a general model, and a summary of the theory of arithmetic hierarchy has been provided by Girard (1987).

2.3.4 – Bennett's "Logical Depth"

Bennett (1988) defines "logical depth" as the execution time required to generate the object in question by a near-incompressible universal computer program, i.e., one not itself computable as output of a significantly more concise program. "Logical depth computerizes the Occam's razor paradigm, with programs representing hypotheses, outputs representing phenomena, and considers a hypothesis plausible only if it cannot be reduced to a simpler (more concise) hypothesis. Logically deep objects, in other words, contain internal evidence of having been the result of a long computation or slow-to-simulate dynamical process and could not plausibly have originated otherwise. Logical depth satisfies the slow-growth law by construction" (Bennett, 1990).

To describe the slow-growth law, Bennett refers to the example of a bottle of sterile nutrient solution (Figure 1) which has higher free energy but lower subjective complexity than a bacterial culture it would turn into if inoculated with a single bacterium. Bennett notes that the rapid growth of bacteria following introduction of a seed bacterium is a thermodynamically irreversible process that is vastly more likely than its reverse process, the transformation of bacteria into high-free-energy nutrient. Bennett suggests that "the unlikelihood of a bottle of sterile nutrient transforming itself into bacteria is therefore not a manifestation of the second law (of physics), but rather of a putative new "slow growth" law that complexity, however defined, ought to obey: complexity ought not to increase quickly, except with low probability, but it can increase slowly, e.g. over geological time" (Bennett, 1990).

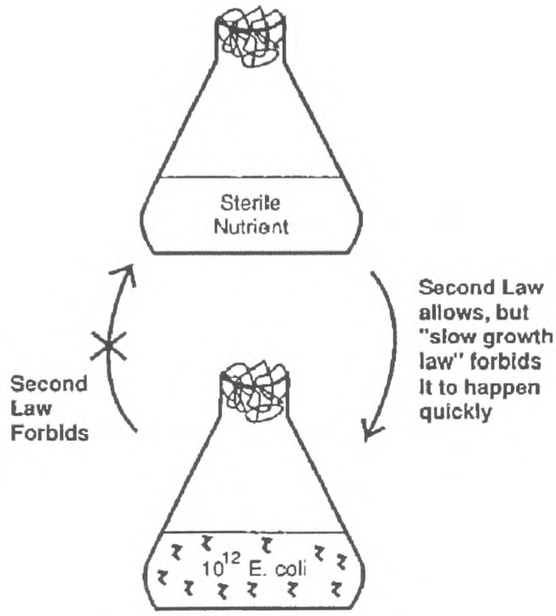


Fig. 1 Bennett's example of the relationship between the second law of physics and the "slow growth law" for turning a bottle of sterile nutrient solution into a bottle of bacteria. For Bennett, complexity is not a thermodynamic potential like free energy. The second law allows the sterile nutrient solution (high free energy, low complexity) to turn into bacteria (lower free energy, higher complexity), but a putative "slow growth law" forbids this to happen quickly, except in the case of low-probability.

2.3.5 - Hinegardner and Engelberg's Number of Parts Definition

Perhaps the simplest measure of complexity is that suggested by Hinegardner and Engelberg (1983): the number of different parts within the system. For Hinegardner and Engelberg, organisms are at root composed of molecules, and without concerning themselves with the composition of and differences between these molecules, they only count the number of parts within the molecules. Smith criticises this measure of Hinegardner and Engelberg by saying that while it gives some indication of complexity, it leaves out what is perhaps most important: "organization" and "levels of organization". In biology, he states, it is essential to distinguish between these different organizational levels and that criteria such as incompressibility (as used in the KCS definition, see Section 2.3.1) or sheer number of parts are not sufficient to describe the differences in complexity between various proteins, for example between haemoglobin protein and insulin (Smith, 1990).

2.3.6 - Descriptive Complexity

In 1969, Fagin (1973) decided to study spectra (the spectrum of a first-order sentence is the set of cardinalities of its finite models) and Asser's problem of whether a class of spectra is closed under complementation (Papadimitriou, 1994). In 1970, Fagin's investigations expanded to generalized spectra (i.e., existential second-order spectra where not all relation symbols are quantified out). Probably Fagin's most important result was his characterization of NP as the class of generalized spectra (1974). Interest in the subject has now exploded, mainly due to the intimate relationship (first hinted at by Fagin) between finite model theory and complexity theory (Banzhaf, 1994). It has now become an established subject area within finite model theory, called descriptive complexity theory.

2.3.7 – Crutchfield's Topological Complexity

The topological complexity described by Crutchfield (1994) is a measure of the size of the minimal computational model (typically a finite automaton of some variety) in the minimal formal language in which it has a finite model. Thus the complexity of the model is "objectified" by considering not only minimal models but also relations to the fixed hierarchy of formal languages.

2.3.8 - Entropy

In physics, entropy measures the level of disorder in a thermodynamic system. The more disordered it is, the more information is needed to describe it precisely. In particular, systems with very low entropy are simple to describe, while systems with high entropy are generally considered more complex (see also Bennett's example in Figure 1 in Section 2.3.4). Thus complexity and entropy can be associated, although this was not intended by its originator (Shannon, 1948). See details on Shannon Entropy in Section 2.3.9.

Entropy based measures are essentially probabilistic. The Boltzman-Gibbs-Shannon Entropy (see Section 2.3.9) is most frequently used in physics, but Algorithmic Information Complexity (see Section 2.3.1) can also be used if the complexity of the whole ensemble is low (Zurek, 1990). In fact, Algorithmic Information Complexity is also often referred to as entropy.

The principle of maximum entropy, as described by Levine and Tribus (1979), has been used to help formalize complexity, as for example described by Cornacchio (1977), Ferdinand (1974) and George (1977). Entropy based complexity measures have often been used for measuring the regularity in noisy time series as described by Pincus (1995), the topology of chemical reactions (Zeigler, 1976), coalitions of economic agents (DeVani, 1993), physical computation (Zurek, 1990), the difficulty of system diagnosis (Golay *et al.*, 1989), artificial life (Ray, 1994), and the complexity of graphs (Moshowitz, 1968).

2.3.9 - Shannon Entropy

The Shannon measure of information, or Shannon Entropy (Shannon, 1998), can be seen as the difficulty of guessing a message passed down a channel given a range of possible messages. It is a statistical measure based on the probability of receiving a message. The idea is that the more difficult it is to guess and the more improbable the message, the more information it gives the recipient. While this was not intended as a measure of complexity, it has been subsequently used as such, especially in the area of Information Theory, a field Shannon founded through his work in the 1940s.

In his seminal paper on "The Mathematical Theory of Communication", Shannon formulated a model of a communication system (Shannon, 1948). He saw the communication process as essentially stochastic in nature. In the Shannon paradigm, information from a "source" (defined as a stochastic process) must be transmitted through a "channel" (defined by a transition probability law relating the channel output to the input). The system designer is allowed to place a device called an "encoder" between the source and the channel, which can introduce a fixed though finite (coding) delay. A "decoder" is then placed at the output of the channel. Shannon then calculates how rapidly or reliably the information from the source can be transmitted over the channel, while optimization of the encoder/decoder side is allowed. He shows that with no loss of generality one can study the source and channel separately and assume that they are connected by a digital (for example binary) interface. One just needs to allow the (source) encoder/decoder to optimize the source-to-digital performance and the (channel) encoder/decoder to optimize the performance of the channel as a transmitter of digital data. Shannon's "source-encoder-channel-decoder-destination model" demonstrates the power of coding with delay in a communication system, the separation of the source and channel coding problems, and the establishment of fundamental natural limits on communication. This discovery has had immense influence on how we currently store, encode and encrypt data, for example on CD-ROMs or storage disks.

In the course of developing this model, Shannon introduced several mathematical concepts, including the notion of the "entropy" of a random variable (and by extension of a random sequence), the "mutual information" between two random variables or sequences, and a calculus that relates these quantities and their derivatives. This technique of random coding shows that an encoder chosen at random from the universe of possible encoders will, with high probability, give essentially optimal performance (Wyner, 1993).

Shannon's information entropy (or entropy rate) is formulated as follows: for a sequence {x} of symbols x drawn from an alphabet * with a probability p(x), the entropy H(X) of the random variable X is given by:

$$H(x) = -\sum_{x \in X} p(x) \log_2 p(x)$$

Fig. 2 Shannon Entropy.

The Shannon Entropy rate can be defined for an arbitrary stochastic process producing successive sequences of symbols. It is a property of a distribution over a discrete set of symbols that is strongly sensitive to the number or variety of the symbols but less so to their relative probabilities of occurrence. The entropy of the sequence has a number of equivalent interpretations and is a measure of the complexity of the random process that generates this sequence. The Shannon Entropy is the length of the shortest binary description of the states of the random variable that generates this sequence, in other words, it is the size of the sequence's most compressed description. Literally, it is the number of binary questions that need to be asked to determine the sequence. It also measures the average surprise, or information gain, resulting from the receipt of a symbol. In other words, the Shannon Entropy measures the complexity or variety of the random variable that underlies a process (Johnston, 1996).

2.3.10 – Goodman's Complexity

Goodman (1966) has devised an elaborate categorisation of extra-logical predicates based on expressiveness. For example, a general predicate is deemed more complex than a symmetric one, as it includes the later as a specific example. Likewise a three-place predicate is more complex than a two place one. Goodman builds upon this starting point. The idea is that when faced with two theories supported by equal experimental evidence, one should choose the simpler one by using this measure.

According to Goodman, the complexity of a complex statement is merely the sum of the complexities of its component predicates, regardless of the structure of the statement.

2.3.11 – Kemeny's Complexity

In the field of "simplicity," Kemeny (1953) attributes an integral measure of complexity to types of extra-logical predicates. He does this on the basis on the logarithm of the number of non-isomorphic finite models that a predicate type has. On the basis of this he gives extra-logical predicates a complexity that could be used to decide between equally supported theories. This is similar in style and direction to Goodman's measure in Section 2.3.10.

2.3.12 - Horn Complexity and Network Complexity

The Horn complexity of a propositional function is the minimum length of a Horn formula (in its working variables) that defines that function. This was used by Aanderaa and Börger (1981) as a measure of the logical complexity of Boolean functions. It is polynomially related to Network or Circuit Complexity, which is the minimum number of logical gates needed to implement a logical function (Savage, 1987).

2.3.13 - Effective Measure Complexity (EMC)

Grassberger (1986) defines the Effective Measure Complexity (EMC) of a pattern as the asymptotic behaviour of the amount of information required to predict the next symbol to the level of granularity. EMC can be seen as the difficulty of predicting the future values of a stationary series, as measured by the size of the required model's regular expression. A similar approach is also taken by Badii and Politti (1997).

2.3.14 - Number of Inequivalent or Non-equivalent Descriptions

If a system can be modelled in many different and irreconcilable ways, then we will always have to settle for an incomplete model of that system. In such circumstances, the system may well exhibit behaviour that would only be predicted by another model. Thus such systems are, in a fundamental way, irreducible. Accordingly, the presence of multiple inequivalent models was considered by Rosen (1977) and Pattee (1977) as the key characteristic of complexity. Casti (1986) extends this approach and defines complexity as the number of non-equivalent descriptions that an observer can generate for a system she interacts with. The observer must choose a family of descriptions of the system and an equivalence relation for them; accordingly, the complexity is the number of equivalence classes the family breaks down into, given the equivalence relation.

2.3.15 - Logical Complexity/Arithmetic Hierarchy

Edmonds (1999) explains that "mathematical proof theorists classify mathematical objects and processes according to the projective hierarchy (sometimes called the Arithmetic Hierarchy). Basically, as one ascends the hierarchy the statements in the classes can have more expressive power and they are more difficult to prove or model (in the mathematical sense)." Girard (1987) gives a good summary of proof theory and logical complexity.

2.3.16 - Loop Complexity

In software metrics various measures are used to predict the maintainability of software, and one these measures is called loop complexity. "The loop complexity of a primitive recursive function is the iteration depth of the primitive recursive register operators in its definition. Thus x+1 would be level 0, x+y level 1 (as it can be defined recursively from x+1), x¥y level 2 etc. This can be used to define a hierarchy of sets *LOOPn* of functions with loop complexity not greater than *n*" (Meyer and

Ritchie, 1967). Zuse provides a good overview of software complexity and the various software metric measures (Zuse, 1991) and (Zuse and Bollmann, 1989).

2.4 - Properties of Complex Systems

Intrinsically linked to defining complexity is the search for properties of complex systems. Various scholars have undertaken the task to define these properties. Again, as for the definitions of complexity (Section 2.3), there is no commonly agreed upon "list" that is thought to completely describe the various complex systems. In the search for complexity criteria that we can apply to the creation of an interactive artwork (see Chapters 7, 8 and 9), we briefly present here a collection of some of the properties that describe or characterize complex systems. For in-depth information on the different complexity measures, we refer the reader to Bruce Edmonds's thesis on "Syntactic Measures of Complexity", a comprehensive collection of the various definitions, measures, properties and characteristics as well as their comparisons (Edmonds, 1999).

2.4.1 - Variety

A complex system is likely to exhibit a greater variety in terms of its behaviour and properties. Thus variety is an indication of complexity (though not always, as sometimes a very complex system is necessary to maintain equilibrium). Variety can be measured by the simple counting of types, the spread of numerical values, or the simple presence of sudden changes.

2.4.2 - Dependency

Heylighen (1996) suggests that complexity increases when the variety (distinction) and dependency (connection) of parts or aspects increase, and this in several dimensions. These include at least the three ordinary spatial dimensions, geometrical structure, the dimension of spatial scale, the dimension of time or dynamics, and the dimension of temporal or dynamical scale. In order to show that complexity has increased overall, it suffices to show that—all things being equal—variety and/or connection have increased in at least one dimension.

2.4.3 - Irreducibility

For holists such as Nelson, irreducibility is considered a source of complexity. Nelson (1976) argues that irreducibility is a key factor in complex systems and similar approaches include the writings by Anderson (1972), who points out the importance of size to qualitative behaviour. Hayek (1964) analyses the number of elements an instance of a pattern must consist to exhibit all the characteristics of a class. Haken (1988) describes a macroscopic approach to self-organization, and Khalil (1995) applies this to the modeling of organizations in nature.

The term irreducibility is also often used in the description of biological systems, and Wombat (1972) argues that the evolution of multiple and overlapping functions will limit reduction in biology. More writings on irreducibility, biological systems, and the holistic approach are provided by Campos (1991) and Yates (1978).

2.4.4 – Minimum Size

The minimum size criteria for complexity seems not a sufficient measure, as it ignores any question of inter-relatedness or relevance. Crutchfield (1993) notes that a system can sometimes be described more efficiently in a different language, and he generalizes the minimal size criterion over the whole formal language hierarchy so that complexity is the minimal size in the "lowest" formal language for which it has a finite description (see also Section 2.3.7. for Crutchfield's Topological Complexity).

Lopez and Caufield (1991) applied the minimum size criteria to minimal complexity in evolution and Wolfram has used the minimum size criteria to measure static complexity of cellular automata (Wolfram, 1984). Note also that when the perfect language is chosen to efficiently model a system that does eliminate any needless length, the minimum size criteria corresponds to the perfect compression principle (incompressibility) as defined in the Algorithmic Information Complexity (see Section 2.3.1).

2.4.5 – Number of Dimensions

If a model has sophisticated relationships between its dimensions and if these dimensions cannot be reduced to composite dimensions, the model is said to have a high potential for complexity. Dimensionality as a measure of complexity has been used by Marcus (1977) for measuring the complexity in networks. In cognitive psychology, Kelly measured the complexity of peoples' personal relationships depending on their dimensionality. In his model, subjects who assign to all their friends positive attributes and negative attributes to their enemies would have a one-dimensional mental model of their acquaintances, since everybody is aligned along this good/friend –bad/enemy scale. On the other hand, subjects who place their relationships across a good-bad, friend-enemy pair of axes would, in Kelly's model, be considered "cognitively more complex". The dimensionality within interpersonal relationships is here seen as a complexity measure (Kelly, 1955).

2.4.6 – Cyclomatic Number

In Graph Theory, a sub-field of discrete mathematics, the inter-connectedness of a graph can be measured. Graph Theory was first studied systematically by D. König in the 1930s (Gardner, 1984).

A graph is a mathematical object composed of points known as vertices or nodes and lines connecting some (possibly empty) subsets of them, known as edges. Formally, a graph is a binary relation for a set of vertices. If this relation is symmetric, the graph is said to be undirected; otherwise, the graph is said to be directed. Graphs in which at most one edge connects any two nodes are said to be simple graphs. Vertices are usually not allowed to be self-connected, but this restriction is sometimes relaxed to allow "loops." The edges of a graph may be assigned specific values or labels, in which case the graph is called a labeled graph. A non-simple graph with no loops but which can contain more than one edge between any two points is called a multigraph. Figure 3 show various possibilities of simple graphs and non-simple graphs. All definitions on graphs are taken from Weisstein (2002).

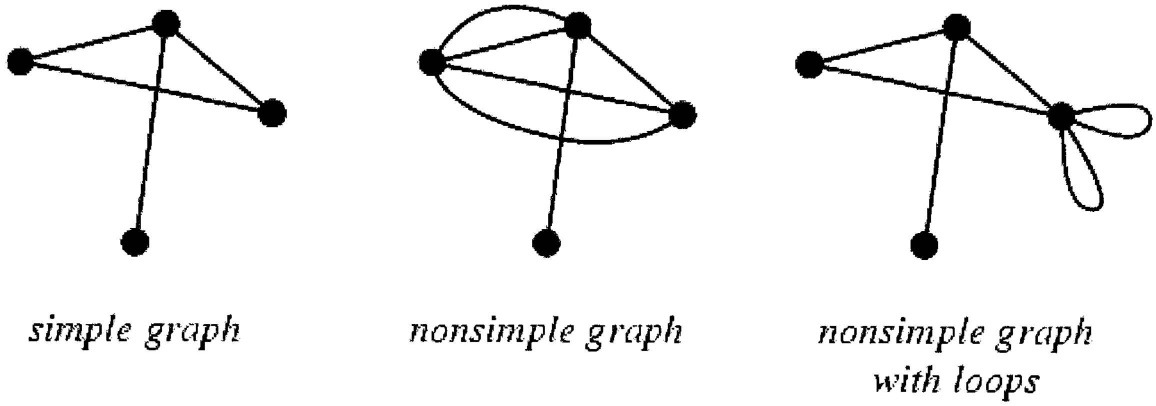


Fig. 3 Simple and Non-simple Graphs.

A degenerate edge of a graph which joins a vertex to itself is called a self-loop. A simple graph cannot contain any loops, but a pseudograph can contain both multiple edges and loops. A pseudo-graph is a non-simple graph in which both loops and multiple edges are permitted. An image of a pseudograph is shown in Figure 4.

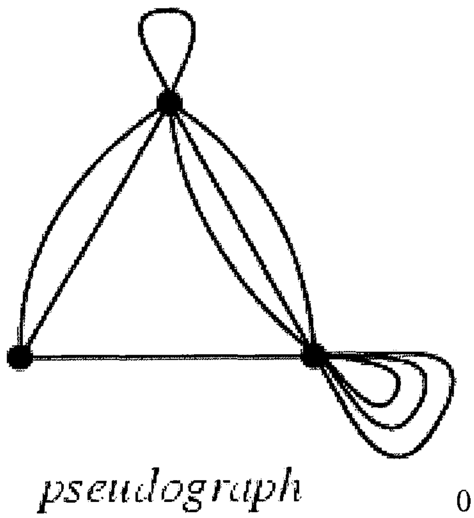


Fig. 4 Pseudograph with multiple edges and loops.

The Cyclomatic Number (also called Circuit Rank) is then the smallest number of edges which must be removed from a graph of N edges and n nodes such that no circuit remains. The formulation of the Cyclomatic Number is as follows:

$$\gamma = N - n + 1.$$

Fig. 5 Cyclomatic Number

In general, there is no direct relation between the size of a graph (or its number of nodes) and its cyclomatic complexity, but the Cyclomatic Number intuitively captures the inter-connectedness of a graph or a system, since a graph with many loops can exhibit more complex behaviour. McCabe (1976) uses graph theory and cyclomatic complexity to measure program complexity and to calculate the number of different logical paths through a program to gauge how many tests it might need. More information on graph theory is available in Gardner (1984), Harary (1994) and Trudeau (1994).

2.4.7 - Ability to Surprise

The ability to surprise is not possessed by very simple and thus well-understood systems, and consequently has come to be viewed as an essential property of complex systems.

2.4.8 - Connectivity

The greater the extent of inter-connections between components of a system, the more difficult it is to decompose the system without changing its behaviour. Thus the connectivity of a system becomes a good indication of the potential for complex behaviour, in particular the likelihood that the system will achieve an equilibrium. The connectivity of a system has been variously measured, including the number of internal relations (Rouse and Rouse, 1979) and the Cyclomatic Number (see Section 2.4.6).

Casti writes about connectivity in relation to the stability in ecosystems (Casti, 1977) and Margalef (1984) measures the diversity in ecosystems depending on connectivity. Connectivity measures have also been applied to circuit design and stability (Winograd, 1963), and Green has used connectivity as a general measure to describe emergent behaviours in biological systems (Green, 1993).

2.4.9 - Symmetry Breaking

Heylighen (1996) argues that complexity can be characterized by a lack of symmetry, or "symmetry breaking", by the fact that no part or aspect of a complex entity can provide sufficient information to actually or statistically predict the properties of the other parts. This again connects to the difficulty of modelling associated with complex systems and also might relate to the incompressibility principle, as defined in Algorithmic Information Complexity (see Section 2.3.1).

2.4.10 - Low Probability

Dawkins argues in his famous book "The Blind Watchmaker" that the probability of a highly organized complex system, such as a watch or an airplane, to assemble itself out of its parts completely by chance and by itself, is very low (Dawkins, 1986). Therefore, complexity is intuitively associated with low probability.

Algorithmic Information Complexity measures (see Section 2.3.1) and entropy measures (see Section 2.3.8) on the other hand suggest a strong connection between complexity and high probability, especially if complexity is generated somewhere between an ordered and a disordered state (see also "Self-Organized Criticality" as described in Sections 2.6 and "Life at the Edge of Chaos" in Section 2.10.). Golay *et al.*, (1989) show how low probability measures can be applied to system diagnosis, in their case for measuring and diagnosing the state of uncertain systems like nuclear reactors and their safety.

2.4.11 – Information Flow and Information Gain

The amount of information a system encodes, or the amount of information needed to describe a system, has a loose relationship to its complexity. There also seems to be a connection between the amount of information and disorder. According to the KCS definition (see Section 2.3.1), disordered patterns hold the most information, and patterns with the maximum amount of information are indistinguishable from random patterns. Information can be measured deterministically using the Algorithmic Information Complexity (Section 2.3.1) or probabilistically by using entropy (Sections 2.3.8 and 2.3.9). Sometimes a combined approach is taken, as proposed by Gell-Mann and Lloyd (1996).

Grassberger (1986) introduced the "Effective Measure Complexity" (Section 2.3.13), which measures the asymptotic increase in information with increased scale (see also Grassberger's example in Figures 6 and 7 in Section 2.3.14), and Atlan (1987) measured the information increase in the evolution of finite automata. Saunders and Ho (1981) considered the increase in complexity and information during evolution, and Bennett (1985) measured the information increase in evolution through logical depth.

2.4.12 – The Whole is More than the Sum of its Parts

According to Kauffman (1995), the pure evolutionary view of nature in the Darwinian sense fails to explain the vast structures of order found in nature. By stressing only natural selection, patterns of spontaneous order cannot be sufficiently described or predicted. In Kauffman's view, this order arises naturally as an "order for free". As a consequence, life is an expected phenomenon deeply rooted in the possibilities of the structures themselves. Kauffman argues that, considering how unlikely it would have been for life to occur by chance, there must be a simpler and more probable underlying principle. He hypothesizes that life is actually a natural property of complex chemical systems and that if the number of different kinds of molecules in a chemical soup passes a certain threshold, a self-sustaining network of reactions—an autocatalytic metabolism—will suddenly appear. It is thus the interaction between these molecules that enables the system to become more complex than the mere sum of its components (Kauffman, 1995). The idea of irreducibility as a source of complexity (as described in Section 2.4.3), which states that the whole of a complex system cannot simply be reduced to a collection of its parts, also accords well with Kauffman's concept of the autocatalytic metabolism.

2.4.13 - Complexity as Relative to the Frame of Reference

For Edmonds (1999), complexity necessarily depends on the language used to model a system. He argues that effective complexity depends on the framework chosen from which to view/model the system of study. The criticality of scale in the modelling of phenomena also leads Badii and Politi (1997) to focus their characterisation of complexity solely on such hierarchical and scaling effects. To demonstrate the effect of judging complexity depending on the reference which is used, Edmonds refers to Grassberger's (1989) experiment as shown in Figures 6 and 7. When a viewer is asked which of three images in Figure 6 appears most ordered and which most complex, this person usually evaluates the left-hand image as being completely ordered while judging the image in the middle as being most complex or most chaotic. The image on the right is usually judged as being disordered but less complex than the middle image.

Fig. 6 Grassberger's example images of complete order, chaos and complete disorder.

Grassberger (1989) then presents the viewer with possible relationships among these images, where the image on the left could be a small pattern included in the middle image and the middle image could be a small pattern included in the right hand image (see Figure 7). Presented with this new knowledge of the relationships among these images, the right-hand side image suddenly appears most complex.

Fig. 7 Grassberger's example of possible diagrammatic inclusions.

This example illustrates the importance of the language used for representing these relationships, and Edmonds (1999) derives from this a general notion of caution that complexity should only be defined relative to its frame of reference.

2.4.14 - Midpoint between Order and Disorder

As we have seen in previous sections, complexity is often posited as a mid-point between order and disorder. Edmonds (1999) notes that the definition of complexity as midpoint between order and disorder depends on the level of representation: what seems complex in one representation may seem ordered or disordered in a representation at a different scale. Crutchfield's Topological Complexity measure (see Section 2.3.7) deals with this problem by suggesting that the complexity of the model should be "objectified" not only by considering minimal (or incompressible) models but also by relating them to a fixed hierarchy of formal languages.

2.5 - Complexity through Phase Transition

Researchers at the Santa Fe Institute in New Mexico, USA have been looking at emergent structures in nature and have called their approach the new science of Complex System Theory or Complex Systems Sciences (see also Chapter 2.1). Stuart

Kauffman is one of the prominent proponents of this new theory. Kauffman (1995) has modelled a hypothetical circuitry of molecules, a Boolean network model, which basically describes the connections and relations between three elements. These elements can switch each other on or off to catalyse or inhibit one of their productions. As a consequence of this collective and interconnected catalysis or closure, more complex molecules are catalysed, which again function as catalysers for even more complex molecules. The networks described by Kauffman's Boolean network model show stability, homeostasis, and the ability to cope with minor modifications when mutated; they are stable as well as flexible. Kaufmann argues that, provided a critical molecular diversity of molecules appears, life can occur as catalytic closure itself crystallizes. The poised state between stability and flexibility is commonly referred to as the "edge of chaos" (see also "Life at the Edge of Chaos" in Section 2.10), and the transition between the areas of simple activity patterns and complex activity patterns are called a "phase transition". An example of a small autocatalytic set that features a phase transition and catalyses more complex molecules, as modelled by Kauffman, is shown in Figure 8.

Other researchers have also analysed this phase transition between order and chaos. Brian Goodwin (1994) describes this transition phase as a kind of biological attractor: "For complex non-linear dynamic systems with rich networks of interacting elements, there is an attractor that lies between a region of chaotic behaviour and one that is "frozen" in the ordered regime, with little spontaneous activity. Then any such system, be it a developing organism, a brain, an insect colony, or an ecosystem will tend to settle dynamically at the edge of chaos. If it moves into the chaotic regime, it will come out again of its own accord; and if it strays too far into the ordered regime it will tend to "melt" back into dynamic fluidity where there is a rich but labile order, one that is inherently unstable and open to change" (Goodwin, 1994).

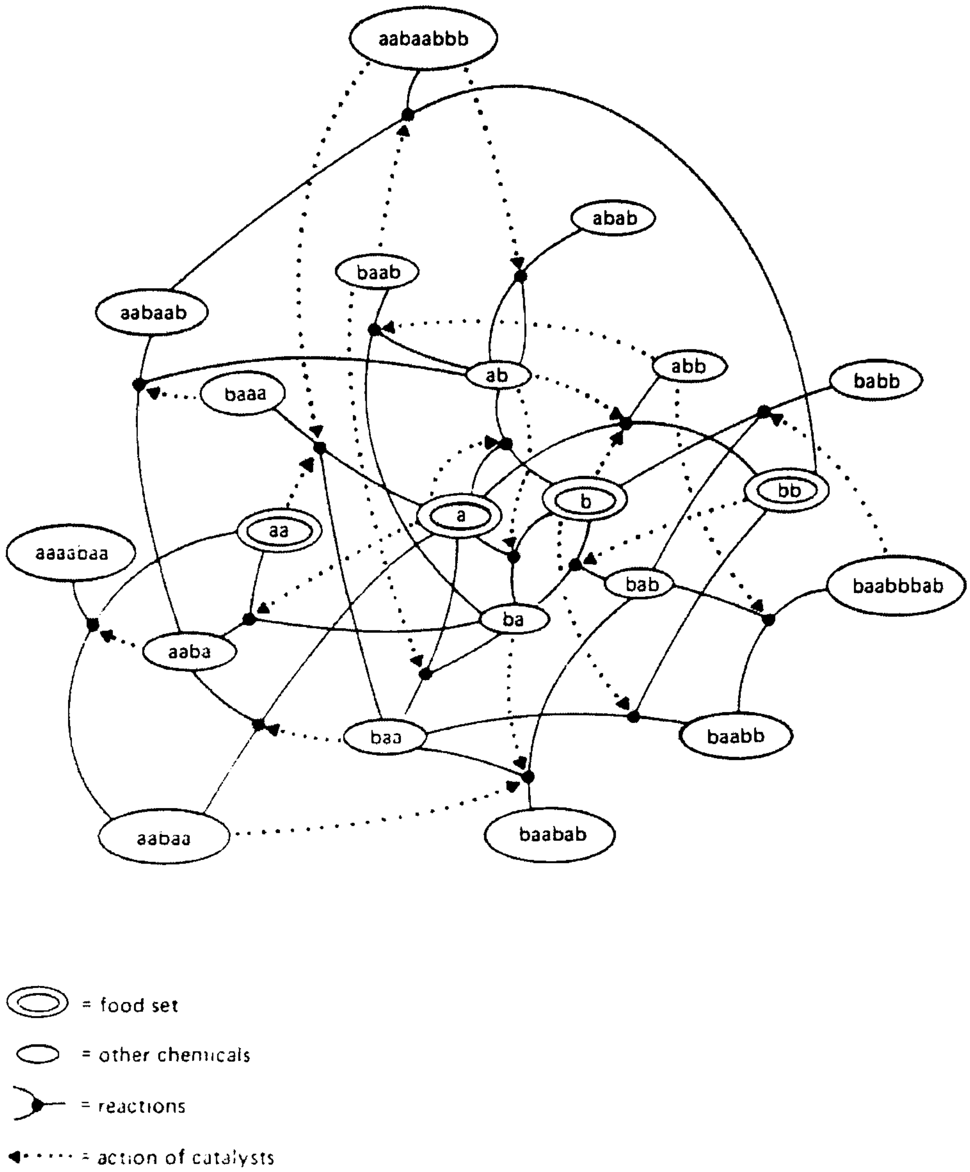


Fig. 8 Kauffman's example of a small autocatalytic set. The reactions are represented by points connecting cleavage products with the larger ligated polymer. Dotted lines indicate catalysis and point from the catalyst to the reaction being catalysed. Monomers and dimers of A and B constitute the maintained food set (double ellipses). Reproduced from (Kauffman, 1993, p. 323).

2.6 - Self-Organised Criticality (SOC)

A related approach involves the models of physicist Per Bak, who sees a connection between the idea of phase transition, or "life at the edge of chaos," and the physical world. In 1987 Bak, Tang and Wiesenfeld coined the term "self-organized criticality" to describe systems which reach a critical state by their intrinsic dynamics, independently of the value of any control parameter (Bak *et al.*, 1987).

The model Bak uses to describe self-organized criticality is a hypothetical sand pile inside an empty sandbox, onto which sand is added at a constant rate. At first, the grains land on the stable slope of a proto-sand pile. As more grains are added, the slope of the pile increases. Eventually, the slope locally reaches a critical value such that the addition of one more grain results in an "avalanche". The avalanche fills in empty areas of the sandbox. With the addition of still more grains the sandbox will overflow. Sand is thus added and lost from the system. When the count of grains added equals the count of grains lost (on average), according to SOC theory, the sand pile has self-organized to a critical state (Bak and Chen, 1991).

The addition of one more grain may or may not result in an avalanche. Eventually, however, an added grain will cause an avalanche. The key elements of this theory are as follows:

1. The next avalanche can be of any size, (ranging from a single grain to a catastrophic collapse of the sand pile). Moreover, the size distribution of the avalanches will follow a power law. For example, if one were to count the number of avalanches and the number of grains involved in each avalanche over a 24-hour period, one would find that there was 1 avalanche which involved 1,000 grains, 10 avalanches which involved 100 grains, 100 avalanches which involved 10 grains, and so on (Figure 9).

2. Simple physical laws dictate the interactions of individual grains of sand. The specifics of these laws are not important, however, as the system will robustly self-

organize to a critical state by a variety of laws. Highly specific physical laws are not necessary in the generation of the power law distribution.

3. Avalanches are not strictly periodic. In other words, although there may be 10 avalanches involving 100 grains in a 24-hour period, one avalanche did not necessarily occur every 2.400 hours.

4. The surface of the sand pile will have a fractal dimension.

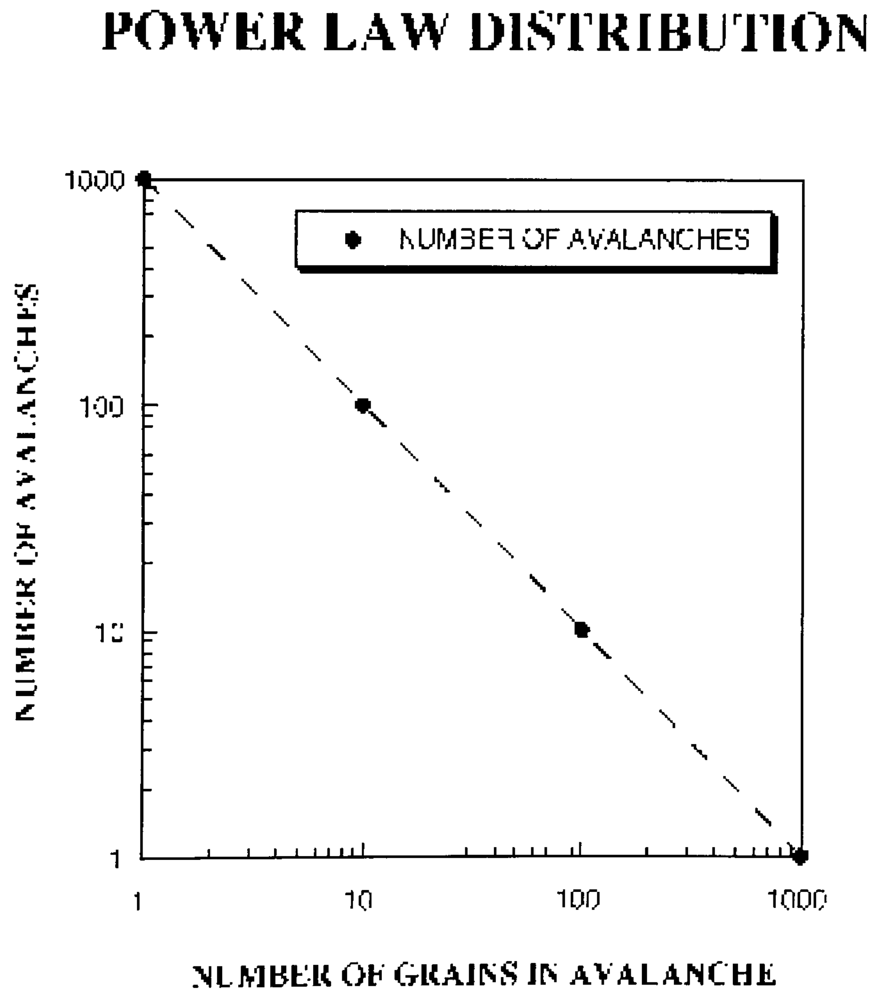


Fig. 9 The power law distribution in Bak's hypothetical sand pile, where the number of grains and the number of avalanches are plotted against each other. The functional relationship is represented as a constant slope, commonly referred to as "self-organized criticality" (SOC).

What makes SOC so intriguing, however, is that it may actually be able to model natural phenomena. The essential link is that these phenomena maintain power law distributions in what can be considered very noisy conditions. In geology, for example, a commonly used example of an SOC is the Gutenberg-Richter magnitude frequency relationship that models seismicity and its subsequent constraints on earthquake rupture physics (Winslow, 97).

To summarize, we can say that SOC is a theory of the internal interactions of large systems and it is hypothesized to link a multitude of complex phenomena observed in nature to simplistic physical laws and/or one underlying process. The essence of SOC states that in large interactive systems self-organization into a critical state occurs as the system is governed by a power law distribution, as illustrated in Figure 9.

2.7 – Prigogine's Entropic Force-Flux Principle (Being-Becoming)

Nobel Price winner and chemist Ilya Prigogine tells us that any dissipative structure in nature (dissipative systems are systems in which entropy is continually being produced and removed to the surroundings, see also Sections 2.3.8 and 2.3.9 on entropy) has a so-called pair where the one becomes the other but will never be the other. This pair is always together but never apart, transitive and asymmetric as well as in recursion. Prigogine calls this an "entropic force-flux pair" or a "being-becoming pair". In physics, this principle is also described by the 1st and 2nd laws of energy conservation (LEC law) and the law of entropy production (LEP law). According to Prigogine, energy is the "being" of the universe, whilst entropy is the "becoming." This "being-becoming pair" can also be described as a "whole" (see Section 2.4.12 for the notion of "the whole is more than the sum of its parts"). The "whole" has an input-transformation-output structure and has content flowing in this structure with a cybernetic link between form (being) and function (content or becoming). Prigogine suggests that most such "being-becoming" processes in nature are time-irreversible, as they are prone to small instabilities and fluctuations, which in return lead to evolutionary patterns that cannot be predicted. In contrast to Newton and Einstein's deterministic view of the universe where time is but an illusion, Prigogine suggests that time is irreversible and life and matter evolve in the direction of time (Prigogine *et al.*, 1997).

2.8 – Parts and Wholes

Researchers and scholars in Complex Systems Sciences are currently discussing the implications of Prigogine's "entropic force-flux pair" as it relates to their search for how the "whole" emerges from its "parts". Complexity specialist Gavin Ritz of New Zealand (Ritz, 2002) suggests in his postings to the on-line complexity mailing list that this "entropic force-flux pair" is probably the most fundamental concept that underlies all Complex Adaptive Systems (see also Sections 2.10 and Chapter 4). One of the challenges for Complex Systems Sciences will be to distinguish which structure-processes (being-becoming pairs) can be considered "wholes", and which cannot.

Some scholars suggest that even complex automated on-line stock trading programs or computer viruses or on-line automated database programs could be considered CASs, as they clearly transform information, learn, adapt and organize, mutate and evolve, react to their neighbours and to external control, organize higher-order structures, explore their options, and even replicate (as defined in Section 2.2). Other scholars participating in the current debate insist, on the other hand, that only beings (structures) that are alive can qualify as "wholes", while non-living matter, such as technical, economic or financial materials or informational structures, can only be considered "quasi wholes", since they always have a human (being) interface to jolt them to action (Ritz, 2002).

2.9 - Emergence

In the study of complex systems, the idea of emergence is used to indicate the arising of patterns, structures, or properties that do not seem adequately explained by referring only to the system's pre-existing components and their interaction. Emergence becomes increasingly important as an explanatory construct when the system is characterized by the following features:

- when the organization of the system, i.e., its global order, appears to be more salient and of a different kind than the components alone;
- when the components can be replaced without an accompanying decommissioning of the whole system;
- when the new global patterns or properties are radically novel with respect to the pre-existing components; thus, the emergent patterns seem to be unpredictable and non-deducible from the components as well as irreducible to those components.

The applicability of emergence as an explanatory construct forms a continuum. At one end, the system can be sufficiently understood by an appeal to the components and their interaction alone; at the other end, an appeal to components and their interactions is simply not very useful in understanding the dynamics of the system as a whole. Because most systems fall somewhere between these two extremes, it is usually not the case that turning to emergence entirely supplants the need to also take into consideration the components and their interactions. Issues involved in using emergence as an explanatory construct include how causality is to be understood in such systems, the question of whether emergence is ever more than a provisional heuristic device to be replaced when there is more knowledge of the components and their interactions, and what general laws or principles can be discerned in the emergent patterns, structures, and properties (quoted from: http://emergence.org/second.htm).

Historically, the concept of emergence was coined by G. H. Lewes (1875) in his "Problems of Life and Mind". C. Lloyd Morgan (1923) wrote that similar concepts are to be found in the theories of J. S. Mill and the psychologist W. Wundt and that these early authors were in agreement regarding the definition: emergence is the denomination of something new which could not be predicted from the elements constituting the preceding condition. In accordance with this, the authors differentiate between "resultants" and "emergents", that is, between properties which can be predicted and properties which cannot be predicted. It is interesting to note that these early definitions of emergence are very close to our modern understanding of the term.

Emmeche (1997) argues that while it is not yet possible to extract a single general theory of emergence, the whole area of complexity science provides an important take-home lesson on emergent structures and that the very idea of emergence should be viewed as one of the most central ideas in modern science. Accordingly, we should aim toward a fully developed theory of emergence, especially in relation to the epistemological and ontological consequences of non-reductionist theories of hierarchical organisation and level theories.

2.10 - Complex Adaptive Systems (CAS) and Life at the Edge of Chaos

Two of the first scientists to describe the emergence of complex patterns within a program of cellular automata and the ones who defined the term "life at the edge of chaos" were Christopher Langton (1992) and Norman Packard. They discovered that in a simulation of cellular automata there exists a transition region that separates the domains of chaos and order. Cellular automata were invented in the 1950s by John Von Neumann (1956). They form a complex dynamical system of squares or cells that can change their inner states from black to white according to the general rules of the system and the states of the neighboring cells. When Langton and Packard observed the behaviour of cellular automata, they found that although the cellular automata obeyed simple rules of interaction of the type described by Stephen

Wolfram (1986), they could still develop complex patterns of activity. As these complex dynamic patterns develop and roam across the entire system, global structures emerge from local activity rules, which is a typical feature of complex systems. Langton and Packard's automata indeed showed some kind of phase transitions between three states. Langton and Packard hypothesized that the third stage of high communication is also the best place for adaptation and change and in fact would be the best place to provide maximum opportunities for the system to evolve dynamic strategies of survival. They furthermore suggested that this stage is an attractor for evolving systems. Subsequently, they called the transition phase of this third stage "life at the edge of chaos" (Langton, 1992).

Other researchers at the Santa Fe Institute have extended this idea of life found in this transition phase and applied it to chemistry. In 1992, Walter Fontana developed a logical calculus that can explore the emergence of catalytic closure in networks of polymers (Fontana, 1992).

In Chapter 4 we will give an in-depth description of Complex Adaptive Systems, their connection to Cellular Automata (CA), and their background in Artificial Life, a research field Christopher Langton founded in 1987.

2.11 - Summary

To summarize, we can say that while there is no exact or unified definition of what a complex system is, several key properties and principles of how such complex systems form have been suggested. They all seem to shared the notions of displaying high entropy (Section 2.3.8), being irreducible (Sections 2.3.1 and 2.4.3), exhibiting high connectivity (Section 2.4.8), existing between order and disorder or at the "edge of chaos" (Section 2.10) or featuring "phase transitions" (Section 2.5), displaying "self-organized criticality" (Section 2.6), and showing an "entropic force-flux pair" that regulates the flow of energy in time (Section 2.7). It also appears that internal changes within these systems can be described by a power law distribution or a change in entropy. These systems are generally assumed to be at the point of maximum computational ability, maximum fitness, and maximum evolvability. Currently agreed upon key characteristics of complex systems, and in particular complex adaptive systems, are that they:

- couple to each other
- learn, adapt and organize
- mutate and evolve
- expand their diversity
- react to their neighbours and to external control
- explore their options
- replicate
- organize a hierarchy of higher-order structures

While this list might not be complete and more properties are currently being discussed on-line (see the Complexity Mailing List, http://necsi.org/ organized by Bar-Yam (2000)), it does however provide an advantageous starting point for creating an artistic system that tries to incorporate some of the features of complex systems and specifically complex adaptive systems. Before introducing some of my own artistic approaches to modelling a complex adaptive system for interactive art (see Chapters 7, 8 and 9), I will briefly look at some related works that served as inspiration and gave me background knowledge to realize these systems. These works are in the fields of Origin of Life Theories (Chapter 3) and Complex Adaptive Systems as related to Artificial Life (Chapter 4). To place my own artwork in the context of other artwork that deals with similar issues, I will also provide a brief overview of existing Artistic Interpretations of Complex Adaptive Systems (Chapter 5) and describe some artworks that deal with other forms of complexity (Chapter 6).

3 – Life's Overall Increase in Complexity and Origin of Life Theories

Directly linked to the modelling of a Complex Adaptive System on a computer are the underlying principles for the emergence of complexity within living systems. Accordingly, it is valuable to study current and common theories on the Origin of Life and to look at the principles and patterns of life's overall increase in complexity, if it indeed increases.

3.1 - Replaying Life's Tape, and Is There an Overall Increase in Complexity?

It is often assumed that complexity increases with evolution. Bedau (1998) argues that the progression of evolution in our biosphere seems to show a remarkable overall increase in complexity, from simple prokaryotic one-celled life to eukaryotic cellular life forms with a nucleus and numerous other cytoplasmic structures, then to life forms composed out of a multiplicity of cells, then to large-bodied vertebrate creatures with sophisticated sensory processing capacities, and ultimately to highly intelligent creatures that use language and develop sophisticated technology. The evidence of this view is consistent with the hypothesis that open-ended evolutionary processes have an inherent, law-like tendency to create creatures with increasingly complicated functional organization. Just as the arrow of entropy in the second law of thermodynamics asserts that the entropy in all physical systems has a general tendency to increase with time (see also Section 2.3.8), the arrow of complexity hypothesis asserts that the complexity in the functional organization of entities produced by open-ended evolutionary systems has a general tendency to increase with time.

Kauffman (1993) explains that by means of self-organisation the complexity in large inter-related systems increases by evolution through the inherent tendency toward order. Dawkins (1989) argues that evolvability itself can evolve, and Wimsatt (1972) suggests that the evolution of multiple purposes for existing internal structures tends to make the workings of an organism more complex. Arthur (1993) proposes that competitive co-evolution of species will result in this increase in complexity.

McShea (1996), on the other hand, calls for caution due to the difficulty of quantitatively verifying this change in complexity. Also, Stephen Jay Gould (1989) expressed scepticism about any global progression of complexity in the evolution of life. He developed the thought experiment of "replaying the tape of life", that is, rewinding the evolutionary process backward in time and then replaying it again forward in time but allowing different accidents, different contingencies to reshape the evolution of life. Gould was confident that "any replay of the tape would lead evolution down a pathway radically different from the road actually taken" in complex systems. A similar view is also shared by Edmonds (1999) and Martinez (1995), who argue that it is unclear why evolution should inherently favour the direction of the more complex rather than the simpler, especially since it is plausible that the simpler is cheaper and easier to maintain.

3.2 - Origin of Life Theories

Speculations on how life on earth originated have a long history, perhaps as long as the history of humanity.

3.2.1 - Rational Morphology and Natural Theology

The search for "laws of form" to explain the patterns of order and complexity seen in nature has intrigued researchers and philosophers since the Age of Enlightenment. These searchers have included famous scholars such as William Bateson (1894), Richard Owen (1861), Hans Driesch (1914), D'Arcy Wentworth Thompson (1942), and Conrad Waddington (1966). Their quest could generally be subsumed under the term Rational Morphology, a counterpart to the functionalistic approach of the Natural Theology promoted by Charles Darwin (1859, 1959) and Neo-Darwinist Richard Dawkins (1986). Whereas Natural Theology considers form mainly a function of natural selection and adaptation, Rational Morphologists emphasize the creative principle of emergence that accounts for the order of structures found in nature. The quest for the "laws of form" is closely linked to the question of the Emergence of Life. The discussion on how life emerged has a long tradition and basically involves two opposing views: the Aristotelian and the Platonic. These two views of the natural world have dominated science over the past two millennia (Lewin, 1993). Baltscheffsky (1997) notes that "Fundamental to a deeper understanding of complex biological functions are ideas about how life originated and evolved. They include questions about how the first compounds, essential to life, appeared on Earth; how the first replicating molecules came into being; how RNA and DNA were formed; how prokaryotes and the earliest eukaryotes emerged; how different species, with traits like susceptibility, sentience, perception, cognition, and self-consciousness, and with various patterns of behaviours, evolved; and how with these developments, the environment and the ecological systems changed."

3.2.2 - Primordial Soup Theory and RNA World Theories

The currently widely accepted hypothesis that life originated from chemical processes largely derives from the work of Russian biochemist Alexander I. Oparin (1924, translated to English, 1938). In the 1930s, Oparin and J.B.S. Haldane (1932) suggested that life on earth could have emerged by natural means in an early atmosphere filled with different gases such as methane, ammonia, hydrogen and water vapour. Oparin and Haldane called this early atmosphere the "Primordial Soup". In their Primordial Soup Theory, life would have originated in the sea as a reaction of these chemical gases triggered by the energy of lightning, ultraviolet radiation, volcanic heat and natural radioactivity.

In the early 1950s, Stanley Miller (1953) of the University of Chicago's Chemistry Department simulated such a primordial atmosphere and was able to synthesize significant amounts of amino acids, the main components of all life forms, from methane, ammonia, water vapour and hydrogen. This experiment gave credence to the belief that the chemical building blocks of life could be created by natural physical processes in the primordial environment. Modern proponents of the Primordial Soup Theory now think that the first living things were random replicators that assembled themselves from components floating around in the primordial soup (Miller, 1953). Based on experiments by Sol Spiegelman (1967), who was able to create self-replicating RNA strings in an environment filled with a primitive "seed" virus and a constant supply of replicase enzymes, Manfred Eigen (1992) went a step further by omitting the initial "seed" virus. Eigen succeeded in showing that self-replicating RNA strands can assemble themselves from only replicase enzymes. In Eigen's theory of the origin of life, RNA molecules can evolve self-replicating patterns and finally develop a primitive genetic code. As the molecules specify and take on different functions, complex and cooperative interactions take place: Eigen calls these the "hypercycles" (Eigen, 1992). Mutation and competition among these hypercycles finally create prototypes of modern cells, and the earlier chemical evolution is finally replaced by biological evolution. A similar theory on the origin of life was also presented by Walter Gilbert (1986).

Even though the "RNA world" model seems very convincing, the question of where RNA came from in the first place remains open. L. Orgel (1987), C. Böhler (1995), and P. Nielsen (1991) found that a peptide nucleic acid, called PNA, could be a pre-form of RNA because it can act to transcribe its detailed genetic information directly to RNA; consequently, PNA could have initiated the RNA world. Another scientist, Hendrik Tiedemann, suggests that the nucleotide bases and sugars needed in RNA could have been built from hydrogen cyanide and formaldehyde, both available in the early atmosphere of the Earth.

3.2.3 - Dual-Origin Theory: Iron-Sulphur, Thioester and PPi World Theories

Completely opposite to the "RNA world" theories on the origin of life is the "Dual-Origin Theory" of A.G. Cairns-Smith (1982). According to Cairns-Smith, the starting point in early crystallization of life was not "high-tech" carbon but "low-tech" silicon, a component of clay. In his theory clay has the capacity to grow and re-assemble itself by exchanging its ion components through mutation and mechanical imperfections. More recent proponents of the mineral and early molecular based theories on the molecular evolution of metabolism subscribe to the "iron-sulphur world" theory of Wächtershäuser (1997), the "thioester world" theory of deDuve (1991), and the "inorganic pyrophosphate world", or "PPi world", theory of Baltscheffsky (1991). Wächtershäuser (1994) proposes a model where early evolution of life as a process begins with chemical necessity and winds up in genetic exchange.

Somewhat related to the question of how life occurred in the first place, whether the first stages of life were metabolic or genetic, is the question of how to draw the line between life and non-life. While generally it is agreed that the RNA world (Gilbert, Eigen, Böhler, Nielsen, Orgel) is a first stage of life, Wächtershäuser (1997) and others believe that rather primitive entities on mineral surfaces can also be called alive; however, he calls them "two-dimensional life". On the other hand, Maynard Smith and Szathmáry (1995) stress that a living organism needs to possess at minimum a reproduction mechanism, and Gánti (1979) proposes that a minimum requirement for a living organism is that it possess three essential subsystems: a genetic system, a functioning unit synthesizing the components, and a membrane part.

Another big question in understanding life's origin is to determine the origin of the translation apparatus and the genetic code (Crick, 1968, Crick *et al.,* 1976, Woese, 1967). Clas Blomberg (1994) claims that the only way to get a stable translation mechanism is a feedback between the code and the proteins that are synthesized by the mechanisms they control. Furthermore, Maynard Smith and Szathmáry (1995) suggest that the relations between amino acids and nucleic acid sequences were established before the translation apparatus, serving as an improved catalyst in the RNA world.

3.2.4 - Other Origin of Life Theories

It would exceed the scope of this thesis to describe all the other theories on the origin of life in detail; however some of them should be mentioned here briefly. These include the "Membrane First" theory of Harold Morowitz (1992) and the "Self-replicating protein" theory of Ghadiri *et al.*, (1996). Theories that life was first introduced by meteorites from other planets or stars include the "Radiopanspermia" theory of Hoyle and Wickramasinghe (1979), the "handedness of the solar system" theory of Carl Chyba (1997) and the "Chirality" theories of Yoshihisa Inoue (1992).

John Casti notes in his book "Paradigms Regained" (Casti, 2000) that "when it comes to defining what it means to be alive, there are as many answers as there are biologists." While the numerous theories about the origin of life suggest that scientists today are still in the dark about the details of life's beginnings and have not been able to create it from scratch, Dawkins (1986) argues that this is rather to be expected. "If the spontaneous origin of life turned out to be a probable enough event to have occurred during a few man-decades in which chemists have done their experiments, then life should have arisen many times on Earth and many times on planets within the radio range of Earth."

3.3 - Connection to Complex Adaptive Systems and Artificial Life

Since the early 1980s researchers have increasingly used computer simulations and computer models to model living systems and to study the emergence of complexity within simulated evolutionary environments. Some of the most prominent examples include the simulations by Dawkins (1986), Reynolds (1987, 1992), Ray (1991), Yaeger (1994), Holland *et al.* (1994), and Langton *et al.* (1995). Called Complex Adaptive Systems, these models typically simulate local interactions between individuals in a group and study how their interactions give rise to novel behavioural patterns. Typically, the resulting global behaviour of the system cannot be explained by simply summing up the local interactions (see also "The Whole is More than the Sum of its Parts" in Section 2.4.12); rather, such a system produces complex patterns such as evolution, self-organization and emergence. Some of these Complex Adaptive Systems and their background in Artificial Life research will be described in the following chapter.

4 - Complex Adaptive Systems and Artificial Life

4.1 - What are Complex Adaptive Systems?

The operational model of the complexity paradigm is a complex adaptive system (CAS). Complex adaptive systems (CAS) consist of many interacting and adapting components. The complexity paradigm uses systemic inquiry to build fuzzy, multivalent, multi-level and multi-disciplinary representations of reality. Systems can be understood by looking for patterns within their complexity, patterns that describe potential evolutions of the system. Descriptions are indeterminate and complimentary, and observer dependent. Systems transition naturally between equilibrium points through environmental adaptation and self-organization; control and order is emergent rather than predetermined (Dooley *et al.*, 1995; Lewin, 1992; Waldrop, 1992).

Although many researchers have studied CAS, there is still no concise nominal definition. An operational model that is currently widely used states, that the interacting and adapting components in a CAS should show the following characteristics. They should:

- learn, adapt and organize
- mutate and evolve
- expand their diversity
- react to their neighbours and to external control
- explore their options
- replicate
- organize a hierarchy of higher-order structures

As we have seen in Section 2.6, Langton and Packard (1992) found that cellular automata not only obey simple rules of interaction like those described by Wolfram (1986) but also have the potential to develop complex patterns of activity. As these complex dynamic patterns develop and roam across the entire system, global structures emerge from local activity rules, which is a typical feature of complex systems. Langton and Packard hypothesize that the third stage of high communication is also the best place for adaptation and change and, moreover, would be the best place to provide maximum opportunities for the system to evolve dynamic strategies of survival and create a complex adaptive system. They further suggest that this stage is an attractor for evolving systems. Subsequently, they called the transition phase of this third stage "life at the edge of chaos" (Langton, 1992).

4.2 - Background of Complex Adaptive Systems in Artificial Life

To understand Complex Adaptive Systems, one has to first study and consider its basis in Artificial Life. Artificial Life (A-Life) was established as a field of research in 1987 by Christopher Langton at the first Artificial Life Workshop in Los Alamos, New Mexico, USA. One of the central ideas of Artificial Life research is to use informational concepts and computer modeling to study life in general and terrestrial life in particular (Boden, 1996). As Langton put it, the quest is not so much to discover "life-as-it-is" but to investigate new possible forms of life, or "life-as-it-could-be" (Langton, 1989). Instead of analyzing life *in vitro*, A-Life researchers attempt to synthesize life *in silico* by using computer technology and its inherent information structure as the basis for creating virtual life.

Other central concepts in A-Life research include self-organization, the emergence of order and complexity from simpler subsystems, and the non-linearity and local determination of behaviour; this means that the properties of the interaction between the parts of a system are of primary interest rather than the inherent properties of the parts themselves. Commonly accepted key properties of Artificial Life system include:

- spontaneous generation of order
- self-organisation and cooperation
- self-reproduction
- metabolization
- learning
- adaptation
- evolution

When we compare these properties with the properties of CAS (as shown in Section 4.1) the strong similarity and connection between Artificial Life and CAS becomes apparent.

Artificial Life is a highly interdisciplinary field of research drawing from the life sciences (theoretical biology, biochemistry, genetics, ecology, physiology, psychological sciences, artificial intelligence, cognitive sciences, linguistics, etc.) as well as from the humanities (economics, the arts, sociology, etc.) in an attempt to understand the essential characteristic processes of living systems. These processes include self-organization, metabolization, self-reproduction, and adaptive evolution (Bedau, 1996).

4.2.1 - Philosophy of Artificial Life (A-Life)

The philosophical issues raised by Artificial Life include basic metaphysical questions about the fundamental aspects of reality, such as life, mind, and emergent phenomena in general. One example of the impact of Artificial Life on the philosophy is the light it sheds on the perennial philosophical question of the nature of emergent phenomena in general. A second example is the way it highlights and promises to explain the suppleness of mental processes. Artificial Life's computational thought experiments also provide philosophy with a methodological innovation (Bedau, 1998).

In the attempt to capture the simple essence of vital processes, Artificial Life models are created as only abstract representations of natural living systems, not as accurate models of particular features of particular natural systems (Bedau 1995). These are "idea" models for exploring the consequences of certain simple premises. Artificial Life simulations are in effect thought experiments, but emergent thought experiments. Philosophers have welcomed new kinds of evidence into their discussions. Details about the contingencies of neurophysiology, for example, inform work on the philosophy of mind (P. S. Churchland, 1986 and P. M. Churchland, 1989), and treatments of reductionism in biology, to pick another example, advert to detailed discoveries of biological science (Kitcher, 1984 and Waters, 1990). We now also find Artificial Life's computer simulations being imported into philosophy. But what is distinctive about Artificial Life's impact on philosophy is that its computational methodology is such a direct and natural extension of philosophy's traditional methodology of *a priori* thought experiments (Bedau, 1998).

Dennett (1995) assumes that evolution by natural selection can explain human concerns like mind, language, and morals, and he points out that Artificial Life can be conceived as a special sort of philosophy which allows the creation and testing of complex thought experiments "kept honest by requirements that could never be imposed on the naked mind of a human thinker alone". To Dennett (1994) Artificial Life consists basically in the creation of prosthetically controlled thought experiments of indefinite complexity. By increasing the capacity and precision of the human mind through computers as prostheses, we would be prepared to increase indefinitely the complexity of such experiments as well.

Moreno (2000) adds that the field of Artificial Life research opens up radically new perspectives not only because it profoundly changes the meaning of concepts like experimentation, modeling and evaluation of theories but also because the philosophy of biology is clearly affected by and highly integrated with technology. All this puts at stake the classical differences made between science and philosophy and demands a more elaborate framework to give proper account of the increasingly complex and dialectical bonds between them. According to Moreno, in Artificial Life the processes of construction and interpretation are deeply entwined, since the essence of an Artificial Life research program is a continuous concatenation of constructions, interpretations, modifications, new interpretations and re-constructions.

4.2.2 - Strong and Weak A-Life

It has been noted the artificial life research community breaks into two clear branches along the lines of strong artificial life (A-Life) and weak artificial life (A-Life). Whereas "weak A-Life" considers that models only represent certain aspects of living phenomena, "strong A-Life" would defend the idea that the phenomenology taking place in the actual computational environment is life in its proper sense.

Maturana and Varela (1980) define life as a type of self-organization: auto-poiesis in the physical space. Boden (2000), who refers to Maturana and Varela (1980), notes that this resembles the concept of metabolism, which itself is typically included in definitions of life. Furthermore, she argues that if life really depends on either auto-poiesis or metabolism (as commonly described in the Artificial Life literature), then the theory of auto-poiesis challenges familiar concepts in biology and cognitive science. Consequently, she concludes that "strong A-Life," which claims that computational simulations of living systems may actually be called living systems, becomes impossible. Boden (2000) further argues that while Artificial Life's use of informational language is too restrictive, its use of cognitive language is too liberal: according to her, life does not imply cognition.

Similar critical remarks on "strong A-Life" claims come from Emmeche (1992), who thinks "strong A-Life" seems to be counterintuitive from a biological point of view. He says that hardly any biologist would disagree with Langton's statement that "Neither nucleotides nor amino acids nor any other carbon-chain molecule is alive—yet put them together in the right way, and the dynamic behaviour that emerges out of their interactions is what we call life" (Langton, 1989). But Emmeche also alludes to Langton's claim that "Life is a property of form, not matter, a result of the organization of matter rather than something that inheres in the matter itself" (Langton, 1989). Though this is a purely philosophical claim rather than a scientific proposition, it appears to be incompatible with the intuition engendered by the current paradigm of molecular biology. This intuition says that real life is both form and matter, and that the proper object of life science is to study both aspects and their dynamic interdependence.

4.2.3 - Cellular Automata (CA)

In the 1930s, Alan Turing pioneered computer science and the design of working computers, the so-called Turing machines. Turing machines are theoretically defined computing systems with an infinite tape, capable in principle of performing any possible computation (Turing, 1936). While Turing himself could not explore the implications of his theoretical research in practice, John von Neumann designed the first digital computer in the late 1940s.

It was also John von Neumann who introduced the concept of cellular automata (CA), the first computational approach to generating life-like behaviour and studying the "logic" of reproduction; in his concept, a cellular automata is a computational "space" made up of many cells. A set of rules (i.e., state transition table) is applied to all of the cells in a grid, each of which can change its state in response to the states of the neighbouring cells (von Neumann, 1966).

Von Neumann and Ulan (von Neumann, 1963) originally introduced cellular automata (CA) as a model of biological self-reproduction. They wanted to know if it would be possible for an abstract machine to reproduce, that is, to automatically construct a copy of itself. Their model consisted of a two-dimensional grid of cells, where each cell had a number of states representing the components used to build the self-reproducing machine. Controlled completely by a set of rules designed by its creators, the machine would extend an arm into a virgin portion of the grid and then slowly scan it back and forth, creating a copy of itself by reproducing the patterns of cells at another location in the grid. Despite their simple construction, some cellular automata are capable of complex behaviour. Figure 10 illustrates a very simple CA which consists of an array of 12 cells, where each cell can have a value of either 0 or 1, represented by the colours white or black, respectively. An example of a CA rule in action is shown on the right side of the figure.

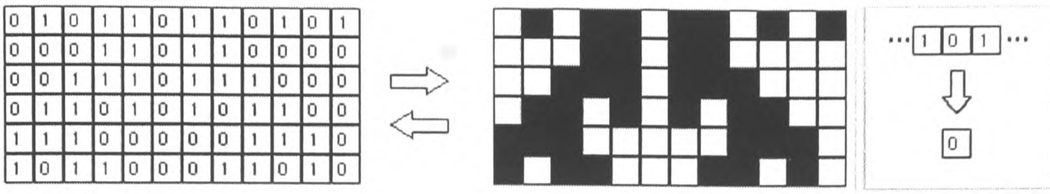


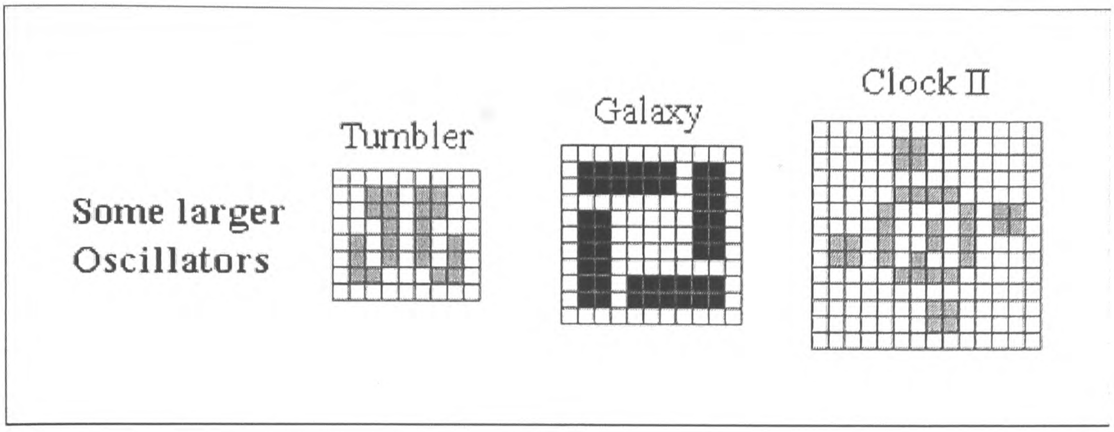
Fig. 10 A simple CA.

4.2.4 - Conway's Game of Life

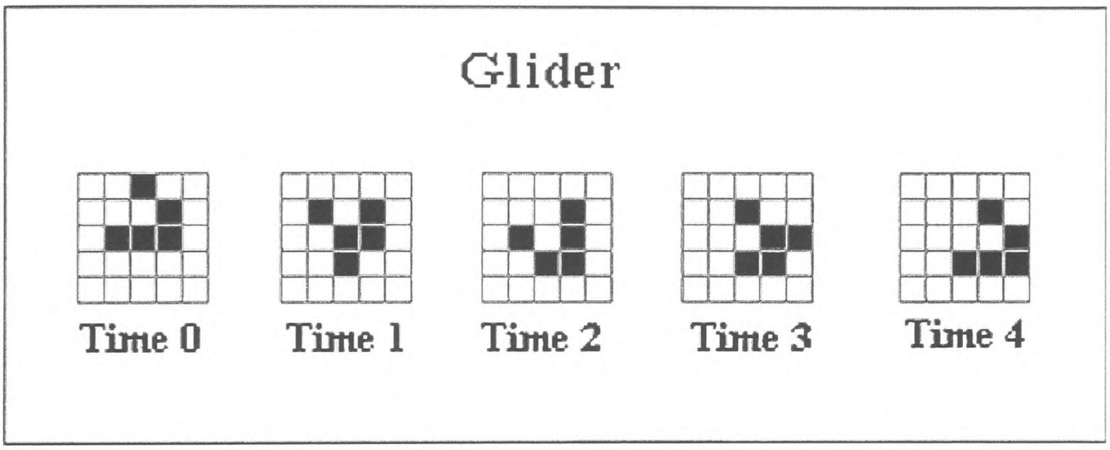
In 1970, John Horton Conway, as described in Gardner (1970), modeled a cellular automaton with only two states instead of the 29 states in Von Neumann's model. These two states are either on or off, depending on the state of the neighboring cells in the grid. From one tick of the clock to the next, the cells of the Game of Life cellular automaton can be either alive (black) or dead (white) according to the following rules:

> GROWTH: if a cell is dead at time t, it comes alive at time t+1 if it has exactly 3 neighbours alive
>
> DEATH: if a cell is alive at time t, it comes dead at time t+1 if it has fewer than 2 or more than 3 neighbours alive
>
> STASIS: if the number of alive neighbours is exactly two, the cell maintains its status quo at time t+1. If the cell is on, it stays on, if it is off, it stays off.

These three simple rules are applied simultaneously to all cells in the grid. An initial configuration of live cells may either grow interminably, fall into cyclic patterns, or eventually die off. Conway discovered that certain patterns emerged with this system, such as gliders, oscillators and blinkers. With "Game of Live", Conway thus proved that simple rules could have complex consequences, where patterns emerge that were not determined by the system beforehand. Figure 11 shows examples of various oscillators and gliders.



(a)



(b)

Fig. 11 Oscillators (a) and gliders (b) in Conway's Game of Life.

4.2.5 - Genetic Algorithms

In traditional evolutionary biology, researchers investigate the *results* of evolution itself (Wilson, 1992). On the other hand, Artificial Life research concentrates on the *process* of artificial evolution. In the spirit of von Neumann's research on the self-reproduction of automata, researchers in Artificial Life began to study the possibilities of artificial evolution. In 1975, John Holland created the concept of "Genetic Algorithms" (GAs), which abstract and apply principles of biological evolution to computer programming. These are search algorithms based on the mechanics of natural selection and natural genetics (Holland, 1992). Similar evolutionary strategies had already been used in engineering design and optimisation since the early 1960s, developed under the names of Evolution Strategies (Rechenberg, 1965, Schwefel *et al.*, 1995) and Evolutionary Programming (Fogel *et al.*, 1975).

A genetic algorithm simulates the evolution of a system by using a Darwinian "survival of the fittest" strategy. There are many variations of genetic (or evolutionary) algorithms. One of the simplest uses a population of bitstrings (a string of 0s and 1s) called "chromosomes" (analogous to molecular biology) to code the solutions to a problem. Each bitstring chromosome can be decoded and applied to the problem at hand. The quality of the solution specified by the chromosome is measured and given a numerical score, called its "fitness", and each member of the population of competing chromosomes is ranked according to its fitness. Low scoring chromosomes are eliminated. High scoring chromosomes have copies made of them (their "children" in the next "generation"). Hence only the fittest survive.

Random changes are made to the children, called "mutations". In most cases, mutations cause the fitness of a mutated chromosome to decrease, but occasionally the fitness increases, making the child chromosome fitter than its parent (or parents, if two parents combine bits "sexually" to produce the child's chromosome). This fitter child chromosome will eventually force its less fit parents out of the population in future generations, until it in turn is forced out by its fitter offspring or the fitter offspring of other parents. After hundreds of generations of this "test, select, copy, mutate" cycle, systems can be evolved quite successfully to perform according to the desired fitness specification. GAs thus efficiently exploit historical information to speculate on new search points with expected improvement in performance (Goldberg, 1989).

4.2.6 - Genetic Programming

In the early 1990s, John Koza (1991) developed a newer version of GAs, called the Genetic Programming Paradigm (GPP), to work in a standard programming language environment. In Genetic Programming (GP), the solutions are defined by trees of lisp-like expressions. The genetic operations of mutation and cross-over can operate at any node of the tree. In the case of cross-over, a node is chosen at random in (typically) two different trees. The nodes, and all of their higher branches and leaves, are simply swapped between the trees.

In this method, the form of the solution does not have to be defined in advance, and so it can also evolve. Although GP exhibits a relatively free-form solution space, it shares with GA the total control of the "fitness function" and the process of replication. By permitting evolution to determine the form of the solution, GP allows a more creative use of the process of evolution, and thus it has been applied to a wide array of problems.

To summarise, we can say that GAs and GPs are computationally simple yet powerful in their search for improvement. They are inspired by the genetic mutation and cross-over functions found in nature and apply the following basic genetic operations.

- Reproduction: often implemented in the form of fitness proportionate reproduction
- Mutation: one or more characters in the string are replaced with another character picked at random
- Cross-over: the analogue of sexual recombination, where two parent strings are picked and lined up, and the sub-strings between them are interchanged to produce two new sub-strings that contain a mix of the parent's genetic information; this operation is extremely useful as it allows an 'intelligent' search and combines parts of the genetic strings that have shown themselves to be useful in combination
- Inversion: used to rearrange the relative locations of certain parts of genetic information within the string of the GA
- Duplication: used when the genome should grow in length

4.3 – Models of Complex Adaptive Systems

To better illustrate what researchers mean when they discuss Complex Adaptive Systems, some of the more prominent models of artificial life and complex adaptive systems are introduced below. Due to space limitations, we are not able to include all models of complex adaptive systems but concentrate on the models that are historically relevant and that have also inspired the artists and designers described in Chapter 5.

4.3.1 – The BIOMORPHS Model

In response to eighteenth-century theologian William Paley's creationist theory which states that the universe is like a watch, too complicated and too functional to have sprung into existence by accident, evolutionary biologist Richard Dawkins argues in his famous book "The Blind Watchmaker" that Paley's analogy is false. According to Dawkins, natural selection has no purpose in mind and that if it plays the role of a watchmaker in nature, it must be a blind watchmaker because it does not see ahead, does not plan consequences, and has no purpose in view. Yet the living results of natural selection overwhelmingly impress us with the appearance of design, as if by a master watchmaker (Dawkins, 1986).

According to Dawkins and based on Darwin's theory, living things come into existence by gradual, step-by-step transformations from simple beginnings, from primordial entities sufficiently simple to have come into existence by chance. Each successive change in the gradual evolutionary process is simple enough, relative to its predecessor, to have arisen by chance. But the entire sequence of cumulative steps constitutes anything but a chance process when one considers the complexity of the final end-product relative to the original starting point.

To illustrate his arguments, Dawkins developed his BIOMORPHS evolution simulation software (Dawkins, 1986). This software consists of randomly generated morphs or abstract line drawings, which one can interactively mate and mutate. When selecting two individuals (mother and father), a generation of off-spring can be created and the children's genome is a combination of the genome of both parents. All children are different from each other, due to the cross-over of the genomes of the two parents and the mutation(s) of their own genome. By selecting two new parents and reproducing them, new generations of morphs evolve, resembling forms found in nature such as ants, flies, fish, frogs, spiders and even trees. Each morph has one chromosome, composed of nine genes. These genes are represented by letters which can take any combination from A to M, except the last which can have only the value from I to N; in total, 4,894,384,326 types of morphs can be displayed. Users can also clone morphs instead of reproducing them, which adds small mutations to each clone, and the parameters for mutations (none, low, medium or high) can be adjusted as well. Figure 12 shows a screenshot of the BIOMORPHS software and its design functions, and Figure 13 shows a collection of biomorphs evolved with this software.

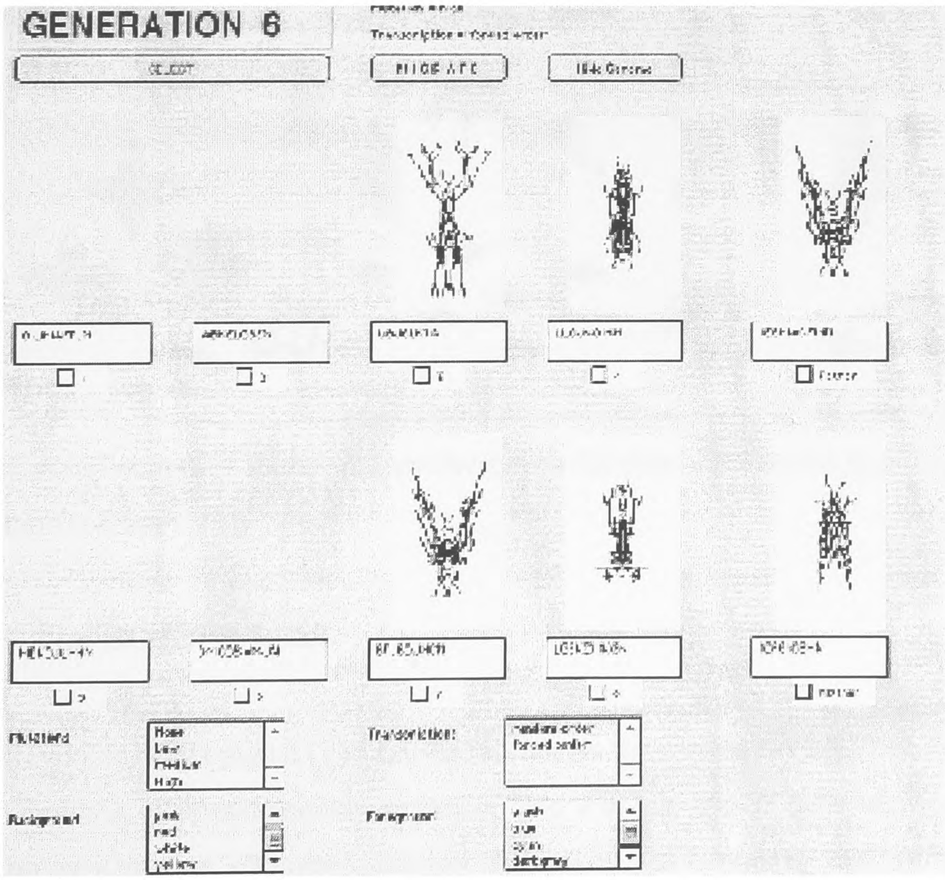


Fig. 12 BIOMORPHS software and design functions.

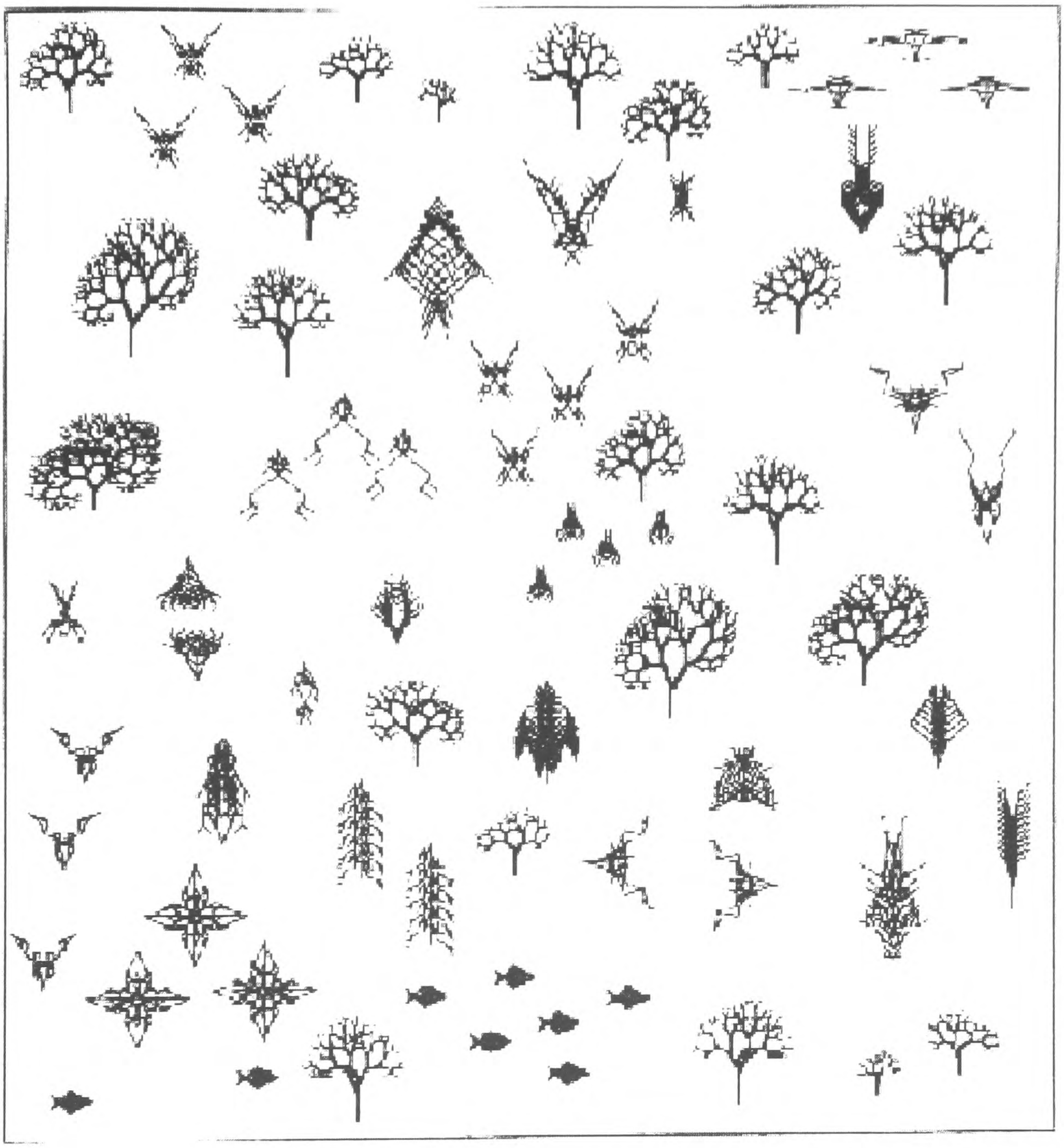


Fig. 13 A collection of biomorphs evolved with Dawkin's BIOMORPHS evolution simulation software.

4.3.2 - The BOIDS Model

Craig Reynolds Boids model (1987, 1992) simulates the natural flocking behaviour of birds ('boids') to spontaneously organize into a flock which then maintains its cohesion as it moves, changes direction and negotiates obstacles, fluidly flowing through space and time. The flock is a loosely formed group, so loose that individual boids sometimes lose contact with the rest of the flock and fly off on their own, only to rejoin the flock if they come close enough to the flock's sphere of influence. The flock appropriately adjusts its spatial configuration and motion in response to internal and external circumstances. The Boids model produces these natural, supple flocking dynamics as the emergent aggregate effect of micro-level boid activity. No entity in the Boids model has any information about the global state of the flock, and no entity controls boid trajectories with global state information. No boid issues flight plans to the other boids. Instead, each individual boid's behaviour is determined by three simple rules that key off of a boid's neighbours:

- seek to maintain a certain minimum distance from nearby boids
- seek to match the speed and direction of nearby boids
- seek to steer toward the centre of gravity of nearby boids

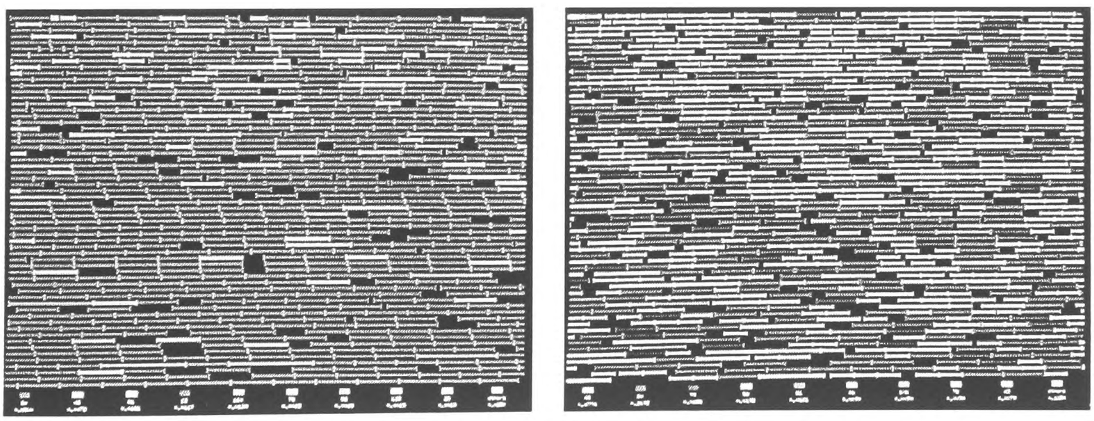
Each boid acts independently in the sense that its behaviour is determined solely by following the imperatives of its own internal rules. (Of course, all boids have the same internal rules, but each boid applies the rules in a way that is sensitive to the contingencies of its own immediate environment.) An individual boid's dynamical behaviour affects and is affected by only certain local features of nearby boids and other nearby objects such as walls and columns. The Boids model contains no explicit directions for flock dynamics. The flocking behaviour produced by the model consists of the aggregated individual boid trajectories, and the flock's global dynamics emerges out of the individual boid's explicit micro-level dynamics. Reynold's Boids model thus provides a good illustration of how complex phenomena of living systems can emerge from simple bottom-up artificial life models. Figure 14 shows an example screenshot of many boids displaying collective behaviour.

Fig. 14 Flocking behaviour of boids in Greg Reynold's Boids model.

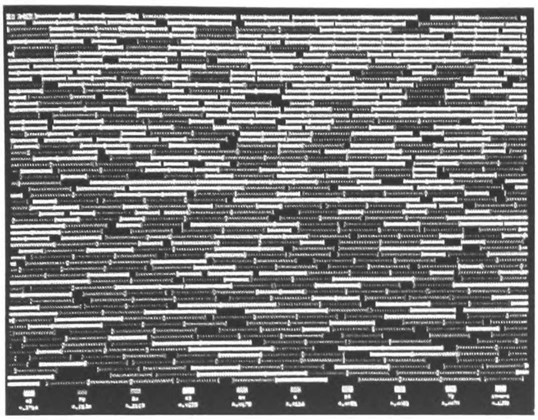
4.3.3 – The TIERRA Model

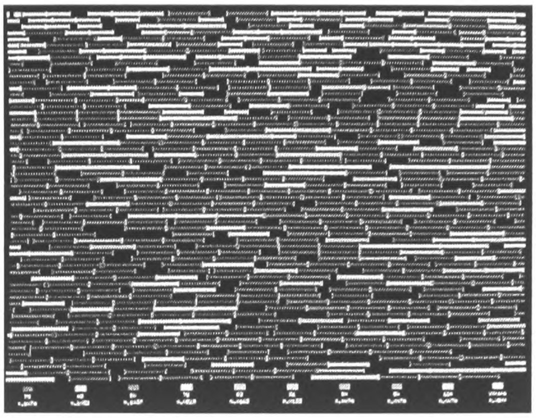
Tom Ray's TIERRA simulator (Ray, 1991) was certainly one of the first systems to clearly investigate artificial evolution in the digital realm. TIERRA is a virtual computer where a block of RAM memory is designated as a "soup" that can be inoculated with artificial creatures. The "genome" of the creatures consists of the sequence of machine instructions that make up the creature's self-replicating algorithm. The prototype creature consists of 80 machine code instructions; therefore, the size of this creature's genome is 80 instructions, and its "genotype" is the specific sequence of those 80 instructions (Ray, 1991). As time unfolds, these creatures start to compete with each other for memory, with which they can start to make copies of themselves. Offspring creatures with new genetic information are created as mutation, and cross-over functions take place within the system. As the digital organisms interact with each other by competing for memory, a complex system evolves that displays many of the real-world characteristics of evolution: parasitism, immunity, hyper-parasitism, sociality and cheating. According to Ray, TIERRA is a system which generates rapidly diversifying communities of self-replicating organisms exhibiting open-ended evolution by natural selection instructions (Ray, 1991).

TIERRA consists of a population of self-replicating machine language programs which "reside" in computer memory consuming the "resource" of CPU time. A TIERRAn "genotype" consists of a specific string type of self-replicating machine code, and each TIERRAn "creature" is a token of a TIERRAn genotype. A simulation starts when the memory is inoculated with a single self-replicating program, the "ancestor", and then left to run on its own. At first, the ancestor and its off-spring repeatedly replicate until the available memory space is teeming with creatures which all share the same ancestral genotype. However, since any given machine language creature eventually dies, and since errors (mutations) sometimes occur when a creature replicates, the population of creatures evolves. Over time, the "ecology" of TIERRAn genotypes becomes remarkably diverse, with the appearance of increasingly fitter genotypes, parasites, and hyper-parasites, among other things. Figure 15 shows the evolutionary race between hosts and parasites in a random soup of TIERRAn creatures.



(a) (b)





(c) (d)

Fig. 15 Evolution in TIERRA. a) Hosts, grey, are very common. Parasites, white, have appeared but are still rare; b) Hosts are now rare because parasites have become very common. Immune hosts, light grey, have appeared but are rare; c) Immune hosts are increasing in frequency, isolating the parasites into the top area of the memory; d) Immune hosts now dominate memory, while parasites and susceptible hosts decline in frequency. The parasites will soon be driven to extinction. Each image represents a soup of 60,000 bytes, displayed as 60 bars of 1000 bytes each. Each individual creature is represented by a greyscale bar, which corresponds to genome size (e.g., grey = 80, white = 45, light grey = 79).

4.3.4 - The POLYWORLD Model

In 1992, Larry Yaeger created a computer-simulated artificial ecology called "PolyWorld" (Yaeger, 1994). In its original conception, "PolyWorld" was targeted principally at the evolution of neural architectures for systems faced with complex behavioral tasks, but it eventually became more focused on modeling ecological dynamics in order to test the emergence of complex behaviors within the system. In this system, a flat world is inhabited by a variety of organisms, represented by simple polygonal shapes as well as some freely growing "food". The artificial organisms can reproduce sexually, fight, kill and eat each other, eat the food that grows throughout the world, and either develop successful strategies for survival or die. An organism's entire behaviour is controlled by the output of its brain. A small number of an organism's neurons are predetermined to activate a suite of possible primitive behaviors, including eating, mating, fighting, moving forward, turning, controlling their field of view, and controlling the brightness of a few of the polygons on their bodies.

4.3.4.1 - PolyWorld's Neural Architecture

Each brain's architecture (it's neural wiring diagram) is determined from its genetic code, in terms of number, size, and composition of neural clusters (excitatory and inhibitory neurons) and the types of connections between those clusters (connection density and topological mapping). Synaptic efficacy is modulated via Hebbian learning (Hebb, 1949) so that, in principle, the organisms have the ability to learn during the course of their lifetimes. The organisms perceive their world through a sense of vision, provided by a computer graphic rendering of the world from each organism's point of view. The organisms' physiologies are also encoded genetically, so both brain and body, and thus all components of behaviour, evolve over multiple generations (Yaeger, 1994).

4.3.4.2 - Metabolism, Predation and Reproduction

Organisms in "PolyWorld" expend energy with each action, including neural activity. They must replenish this energy in order to survive. They may do so by eating the food that grows within the environment. When an organism dies, its carcass turns into food. Because one of the possible primitive behaviors is fighting, organisms can potentially injure other organisms. Consequently, they can also replenish their energy by killing and eating each other. Predation is thus modeled quite naturally. The organisms' simulated physiologies and metabolic rates are determined from an underlying genome, as are their neural architectures. When two spatially overlapping organisms both express their mating behavior, reproduction occurs by taking the genetic material from the two haploid individuals, subjecting it to cross-over and mutation, and then expressing the new genome as a child organism. One way to look at this artificial world is as a somewhat complex energy-balancing problem. The fittest organism will be the one that best learns to replenish its energies by eating and to pass on its genes by mating.

4.3.4.3 - Emergence of Complex Behaviors

Yaeger (1994) observed that from only the simple suite of primitive behaviors within "PolyWorld" certain recurring "species" which display distinctive group behaviours have occurred. These emergent behaviours include responding to visual stimuli by speeding up, responding to an attack by running away, responding to an attack by fighting back, grazing (slowing upon encountering each food patch), expressing attraction to food (seeking out and circling food), and following other organisms. Figure 16 shows a screenshot of "PolyWorld" and its various "species".

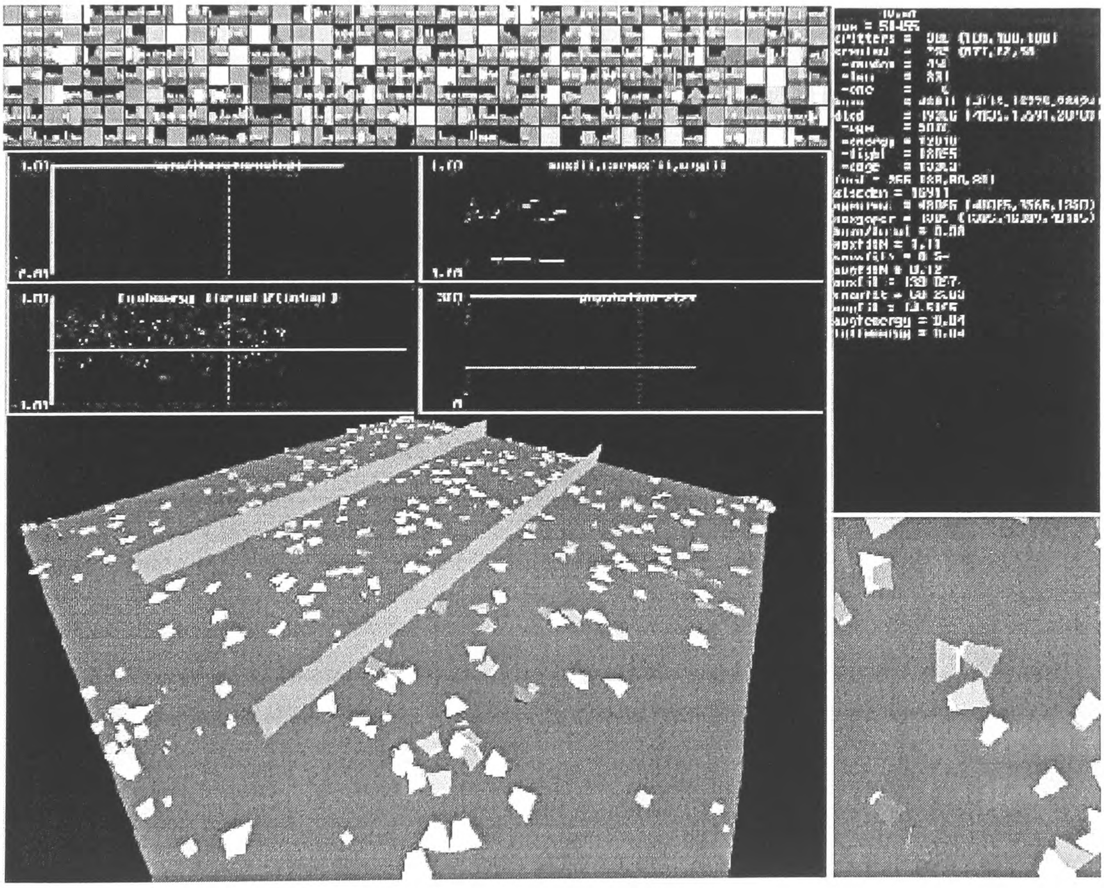


Fig. 16 "PolyWorld" screenshot.

4.3.5 - The ECHO Model

Another evolutionary model created to simulate evolution in the digital realm is the ECHO model by John Holland (1994) and Stephanie Forrest (Forrest *et al.*, 1994). ECHO is a simulation tool developed to investigate the mechanisms which regulate diversity and information processing in systems comprised of many interacting adaptive agents, or complex adaptive systems (CAS). ECHO agents interact via combat, trade and mating and develop strategies to ensure survival in resource-limited environments. Individual genotypes encode rules for interactions. In a typical simulation, populations of these genomes evolve interaction networks that regulate the flow of resources (Mitchell and Forrest, 1997). ECHO thus extends the genetic algorithm concept to model dynamics of adaptive agents in a spatially distributed and resource-constrained setting.

4.3.5.1 - Echo's Tag and Condition Genes

ECHO consists of a collection of agents distributed across a two-dimensional array of sites, resources (represented as letters), and various kinds of interactions, both between pairs of agents and between agents and the environment. Agents have a genome that is roughly analogous to a single chromosome in a haploid species. Tags are genes that produce some externally visible feature of the phenotype, for example color. Conditions are genes that cannot be directly observed by other agents. They encode internal preferences, behavioral rules, and so forth. Thus, one agent will interact with another on the basis of its own internal conditions (rules for interaction) and the other's external tags (appearance). This allows the possibility of sophisticated interactions between agents, including mimicry, bluffing, other forms of deception, and some intransitivities. The six (external) tag and (internal) condition genes possessed by every agent are the offense tag, defense tag, mating tag, combat condition, trade condition, and mating condition. These genes are used to determine what sort of interaction will take place between a pair of agents and what the outcome will be (Hraber *et al.*, 1997)

4.3.5.2 - Replication

An agent is replicated when it acquires sufficient resources. The agent's genome is copied using the resources it has stored in its reservoir to construct the daughter agent. There are two parameters that control the replication process. During replication, there may be spontaneous mutation, the frequency of which is controlled by the mutation probability.

4.3.5.3 - Combat

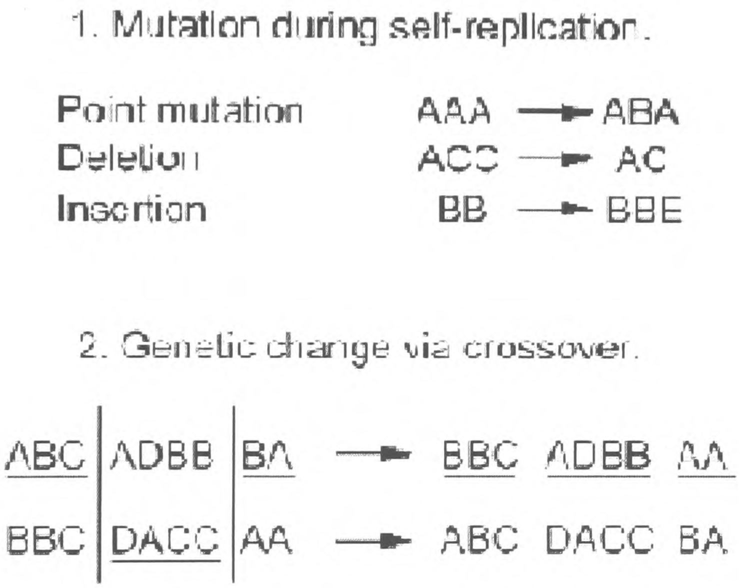
Combat is an idealization of any antagonistic interaction between real-world entities. If two agents in a real-world system were behaving in a competitive fashion, this would be modeled in ECHO by designing the agents that engage in combat. When combat occurs, one agent is killed (and its resources are placed in the reservoir of the survivor), unless it flees first. When two agents meet, the first determination made is whether either agent will attack the other. An agent A will attack an agent B if A's combat condition is a prefix of B's offense tag. If attacked, an agent is given a chance to flee (which it does with a probability equivalent to the probability of it losing in the combat encounter). The calculation of the probability of victory in combat is based on matching A's offense tag with B's defense tag and vice versa.

4.3.5.4 - Trading

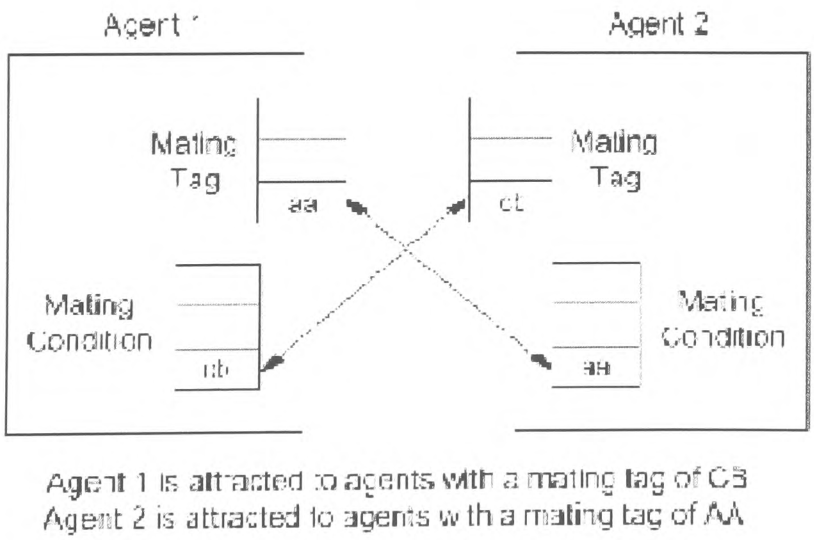
If two agents are chosen to interact and they do not engage in combat, they are given the opportunity to trade and mate. Unlike combat, trading must be by mutual agreement. Agents A and B will trade if A's trading condition is a prefix of B's offense tag and vice versa.

4.3.5.5 - Mating

Agents that interact and do not engage in combat may exchange genetic information through recombination. As in many genetic algorithms, the new agents replace their parents in the population. Recombination occurs between two agents A and B if A finds B acceptable and vice versa. When mating does occur, a form of two-point cross-over is employed. This is complicated by the fact that genomes are of variable length; one can choose a cross-over point in one agent but find that the same cross-over point does not exist in the other. Briefly, cross-over proceeds by (a) selecting two genes to contain cross-over points, (b) choosing cross-over points in each gene in each agent, and (c) crossing over in the manner of two-point cross-over (Hraber *et al.*, 1997). Figure 17 shows an example of operations introducing genetic change in ECHO agents (a) and a simplified view of the two-way tag and condition matching that is used by agents to determine whether mating will occur (b).



(a)



(b)

Fig. 17 a) Genetic exchange in the ECHO model, b) tag and condition matching before mating of ECHO agents.

4.3.6 – The SWARM Model

Another evolutionary simulation project is the SWARM system by Christopher Langton *et al.* (1995). SWARM is a multi-agent software platform for the simulation of complex adaptive systems. In the SWARM system the basic unit of simulation is the swarm, a collection of agents executing a schedule of actions. SWARM supports hierarchical modeling approaches whereby agents can be composed of swarms of other agents in nested structures. SWARM provides object-oriented libraries of reusable components for building models and analyzing, displaying, and controlling experiments on those models. A swarm simply represents a group of agents and their schedule of activity. The modularity and composability of swarms allow for a flexible modeling system. Swarms can be nested to directly represent multi-level simulations or be used by the agents themselves as models of their own world (Langton, Minar and Burkhart, 1995).

The primary feature of Swarm is the virtual machine. The virtual machine allows the researcher to describe agent behaviours one by one, agent-by-agent, context-by-context, all while keeping an exact notion of time and concurrency in the world. Swarm also makes it possible to compose or decompose hierarchies of agents. This attribute is called composability. The primary functionality of a Swarm is to map time from many instances of subjective representation into a single objective representation. In Swarm, physical constituency and temporal vicinity are represented by a "Swarm". Figure 18 shows an example of how the structure of the United States military might map to Swarm. Each box could be considered a Swarm. The diamonds are agents and the circles are aggregate agents. Either can be Swarms or simple objects that make plans through the group to which they belong (the Swarm). A Schedule is an agent's to-do list. There are different kinds of to-do lists and different attributes that Action items on the to-do list can have. An Action is something that happens in the world. In Swarm, Schedules and Actions are typically closely associated with an agent or model component. Agents may have their own Schedules (perhaps several) and a repertoire of Actions they know how to perform. It is also possible for one event to create or remove future events as a side effect. This is called "dynamic scheduling" (e.g. mousetrap) (Daniels, 1999).

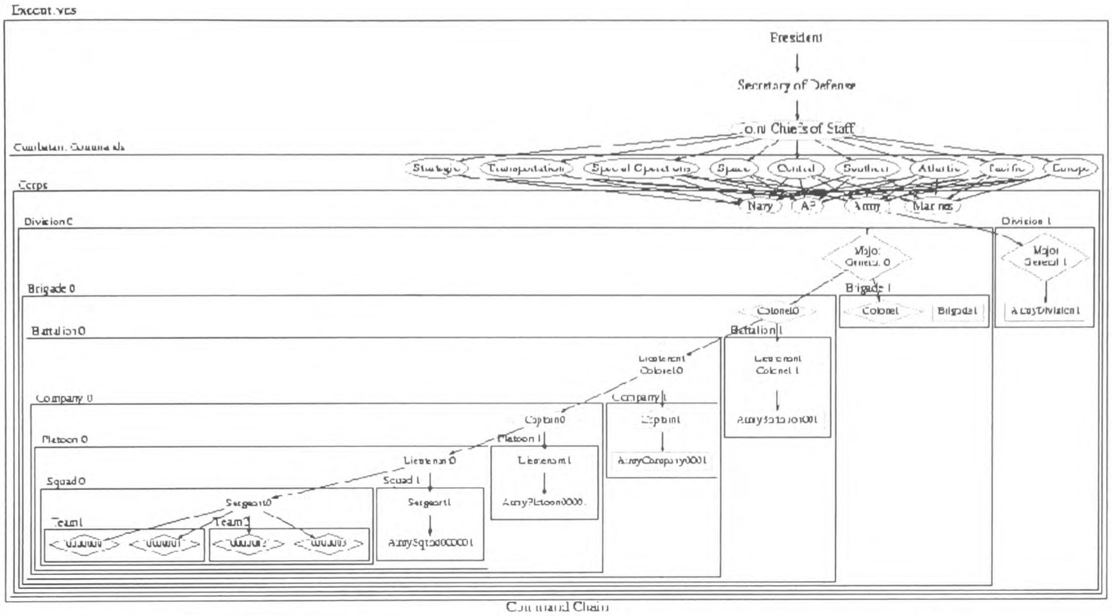


Fig. 18 Example of how United States military could be mapped to Swarm.

4.3.6 - Summary

The true power of evolution lies in its ability to exploit emergent collective phenomena (Langton, 1996). One of the central concepts in Complex Adaptive Systems and Artificial Life research is to create artificial evolution via artificial selection and to test the possibilities of creating self-sustaining, emergent systems with open-ended evolution. As described in Section 4.3.1, the first attempt in this direction was the BIOMORPHS model by Richard Dawkins (1986, 1989) and "The Iterated Prisoner's Dilemma Model" by Lindgren (1991), which showed how co-evolution could take place in a simulated model. Tom Ray's TIERRA simulator (Ray, 1991, 1997) finally showed that artificial evolution can be modeled in the digital realm, and the ECHO model of Stephanie Forrest (1994) and John Holland (1994) showed that populations of individual genomes can evolve interaction networks that regulate the flow of resources (Mitchell and Forrest, 1997) within a systems comprised of many interacting adaptive agents, or complex adaptive systems (CAS)

(Mitchell and Forrest, 1997). In terms of software evolution, Peter Nordin (1998) suggests directly evolving computer code with the Compiling Genetic Programming System (CGPS) by applying GAs to binary machine code while treating the programming language as genetic material. Letters or whole strings in the program can be reproduced, crossed-over, mutated, inversed and duplicated, leading to new and evolved chunks of binary code, or programs, that can be directly executed by the computer's processor (Nordin, 1998).

5 - Artistic Interpretations of Artificial Life and Complex Adaptive Systems

Artists and designers have created artworks that deal with principles and ideas of artificial life, generative systems, and complex adaptive system since the early 1990s. This chapter introduces some of the existing models and separates them into four groups: artworks that deal more-or-less with complex adaptive systems, artworks that use generative musical systems, systems and works in generative design and generative architecture, and finally game products that are based on generative design principles.

5.1 - Overview of Artworks Dealing with Artificial Life and Complex Adaptive Systems

Much inspiration for creating digital life forms on a computer screen and modeling their complex internal behaviours was provided by Dawkin's "Biomorph Land" evolution software, a program in which users can guide the "evolution" of generations of graphical stick figures. This program was published in Dawkin's influential book "The Blind Watchmaker" (Dawkins, 1986). Reynold's subsequent work simulating the flocking behaviour of artificial birds (Reynold, 1987) was another milestone in establishing the idea of applying artificial life principles to computer graphics software. As mentioned earlier, Ray's TIERRA evolution simulator (Ray, 1991) finally brought the possibilities of software evolution to wider attention. But it was the emergence of advanced computer graphics technologies in the early 1990s that helped artists to study the visual creation process itself. In the following sections we will describe various artworks and systems that have been developed by using genetic algorithms (GAs). We will not include artworks that only work with Artificial Life metaphors.

5.1.1 - Latham and Todd's Artificial Life Work

During the late eighties and early nineties, William Latham, in collaboration with programmer Stephen Todd at IBM, created software for synthesizing, mutating and evolving three-dimensional forms (Todd and Latham, 1992). Figure 19 shows examples of these evolving shapes, and the underlying algorithms were later included in their commercial software package "Organic Art," released in 1998. While the early version of their software (1992) was used to produce still images and print-outs for commercial items such as record covers and even T-shirts, the later version of this software (1998) was made interactive, where users could interactively choose and design their own organic scenes. Again, Latham and Todd cleverly commercialized this software, and users could create complex organic 3-D forms which were then mutated and transformed, creating screensavers resembling gems, crystals, creatures, amulets, flowers, DNA, geometric orbits, galaxies, faces and skulls, jewels, metals, and liquid crystals. While this work clearly applied genetic algorithms (as shown in Section 4.2.5) to the creation process of these organic shapes and forms, the main target of the system was not to model a complex adaptive system as defined by the criteria outlined in Section 4.1.

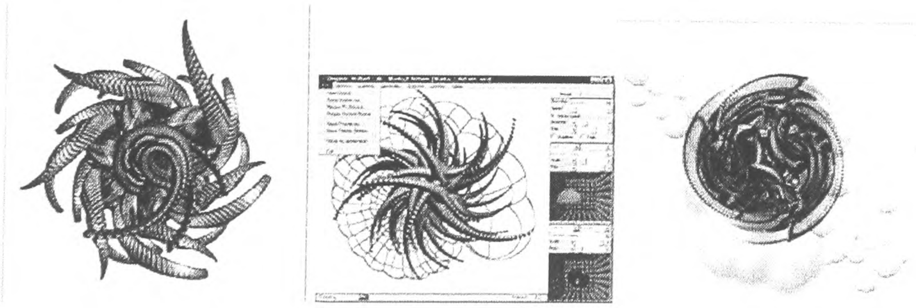
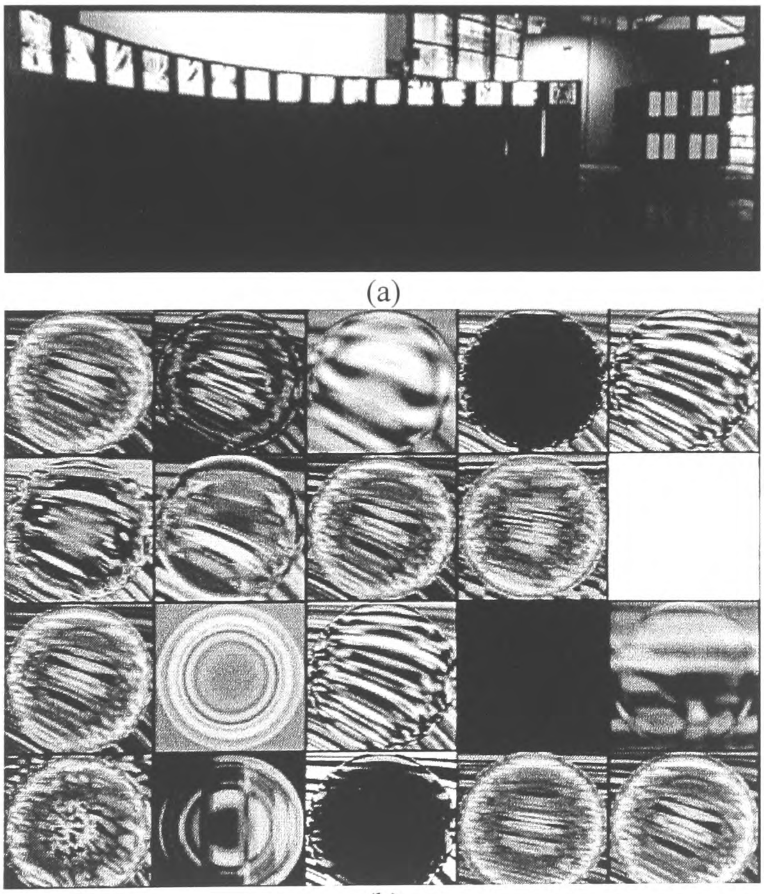


Fig. 19 Examples of evolved shapes generated by Latham's "Organic Art" software.

5.1.2 - Sims's Genetic Images

Perhaps the first artist to apply artificial life principles to interactive computer art was Karl Sims. In his interactive computer installation "Genetic Images", created in 1991, he allowed visitors to choose 2-D images that then would develop by means of genetic crossover and mutation operations. Stepping onto the step sensors in front of the display screens, users in this installation could act as collective "selectors" for generation after generation of evolved images. The images themselves showed remarkable complexity, representing a mixture of human selection and preferences as well as artificial genetics (Sims 1991, 1993a). Figure 20 shows the installation set-up of "Genetic Images."



(a)

(b)

Fig. 20 Sims's "Genetic Images" installation (a) and examples of evolved images (b).

5.1.3 - Sims's Evolving Creatures

In 1994, Sims created a system for the evolution and co-evolution of virtual creatures that compete in physically simulated three-dimensional worlds (Sims, 1994a). The morphology of these creatures and the neural systems for controlling their muscle forces were both genetically determined, and the morphology and behaviour could adapt to each other as they evolved simultaneously (Sims, 1994b). In comparison to Sim's earlier work (Sims 1991), this system did not feature user interaction possibilities, but the system gave a stunning example of artificial learning based on coevolution. Using directed graphs for genotypes, Sims constructed virtual creatures to compete in virtual contests of "capture the flag" and he made videos of their evolved behaviors. Since then, by linking creature evolution with environments supporting artificial physics, other impressive behaviors have been evolved, including artificial walking and swimming. Figure 21 shows a snapshort of several of Sims's Evolving Creatures.

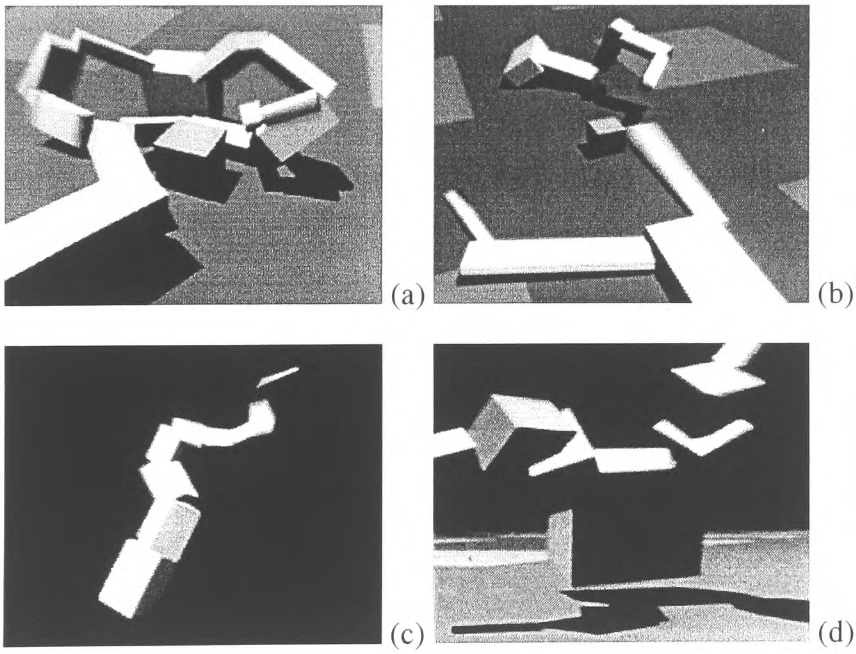


Fig. 21 a)–d) Examples of Sims's Evolving Creatures.

5.1.4 - Sommerer & Mignonneau's A-Volve

In 1994, together with Laurent Mignonneau, I introduced my interactive computer art installation "A-Volve," a pioneering multi-user A-Life system where visitors could actually create artificial life 3-D creatures by drawing 2-D shapes on a touch screen (a section view and a side view) to produce 3-D jellyfish-like forms that started to live and swim in a water-filled glass pool (Sommerer and Mignonneau, 1994). The creature's shape, locomotion and behaviour in this system are solely dependent on its genetic code, which is derived from the section and side view of the user's 2-D drawings. An example of how the side and section view drawings are combined into a 3-D creature is shown in Figure 22.

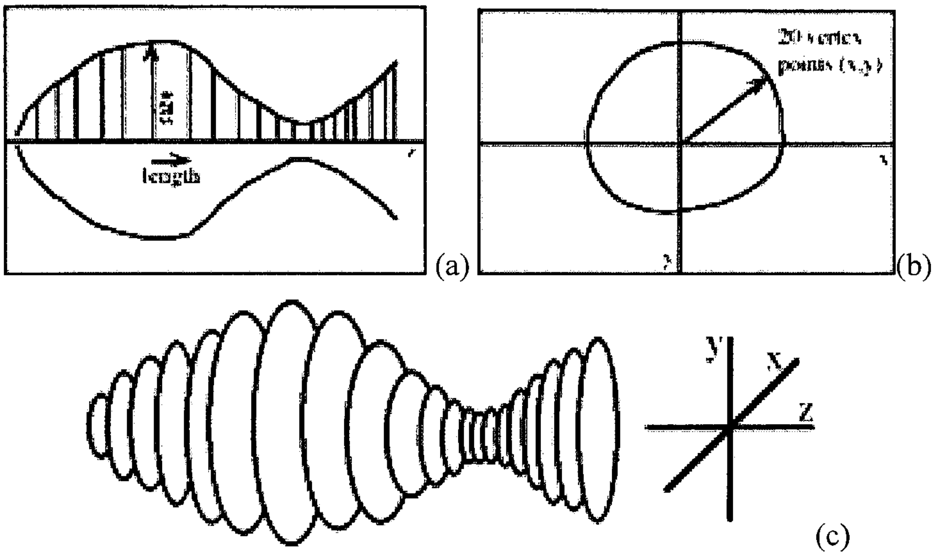


Fig. 22 a)–c) A-Volve creature's side and section view drawings and the resulting 3-D creature.

To derive the genetic code for the creature, the section and side view are each cut into 20 equal sections, providing 20 parameters each for size and length of the side view and 20 parameters each for x and y axes in the section view drawing. This gives us a total of 80 parameters for x, y and z. In addition, colour, texture and brightness values are added at random, using three parameters for colour (RGB red-green-blue), three for brightness (B of RGB), and four for texture (T of RGB and 1 Alpha value). Figure 23 shows the genetic code of the creature with all of its 90 parameters.

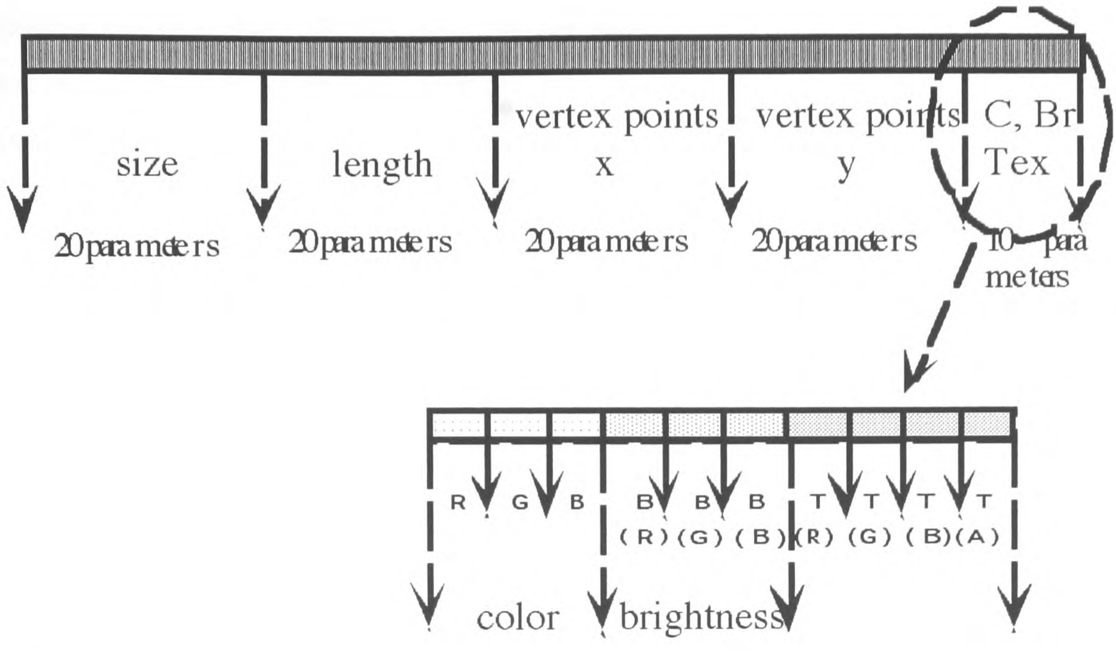


Fig. 23 Genetic code of A-Volve creature created in Figure 22.

A creature's ability to move is determined by its genetic code, which is expressed in its shape and volume. A virtual muscle is placed at the creature's head and it helps to push the creature forward through the virtual water. Depending on its shape, one muscle movement can push the creature more-or-less far. The amount of distance a creature can push itself forward is equivalent to its fitness (F).

Another important parameter that decides the creature's overall behaviour is its energy level (E). This value is E=1 at birth, and when the creature enters the pool it gradually goes down as the creature moves around in the pool. When the critical level of E=0 is reached, the creature becomes hungry and needs to eat. Energy can be added by killing other creatures, that is, "eating up" the energy of a prey creature. Behaviour occurs through the constant balancing and checking of fitness and energy levels between the creatures predator and pray. Weak creatures will be eliminated and fitter creatures survive.

Predator creatures who manage to increase their energy level to $E > 1$ can also become potential parent creatures. If they find a suitable mate with enough energy, mating can take place. In this case, the genetic code of the two parent creatures is exchanged by cross-over and mutation. The resulting child creature is a mix of its parents' genetic code, and it will start to move around in the pool to interact with other creatures. An example of the genetic exchange between two parent creatures is shown in Figure 24.

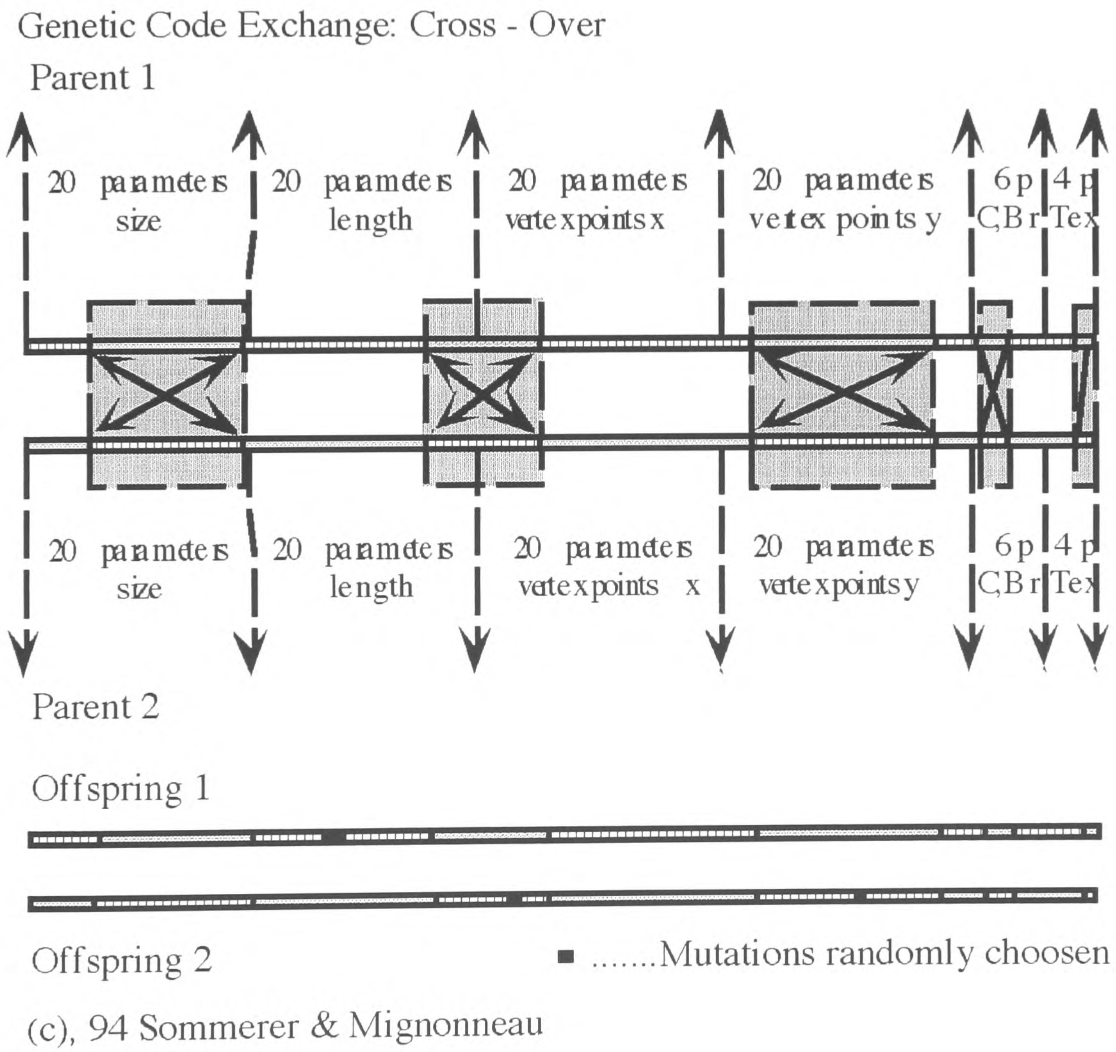


Fig. 24 Genetic exchange between two A-Volve creatures through cross-over and mutation.

Because the genetic code of the offspring is transferred from generation to generation and the system is based on selection of fitter creatures, the entire system is able to evolve over time toward fitter creatures. An overview of the creatures' internal decision parameters and their consequences on the other creatures' behaviours is shown in Figure 25.

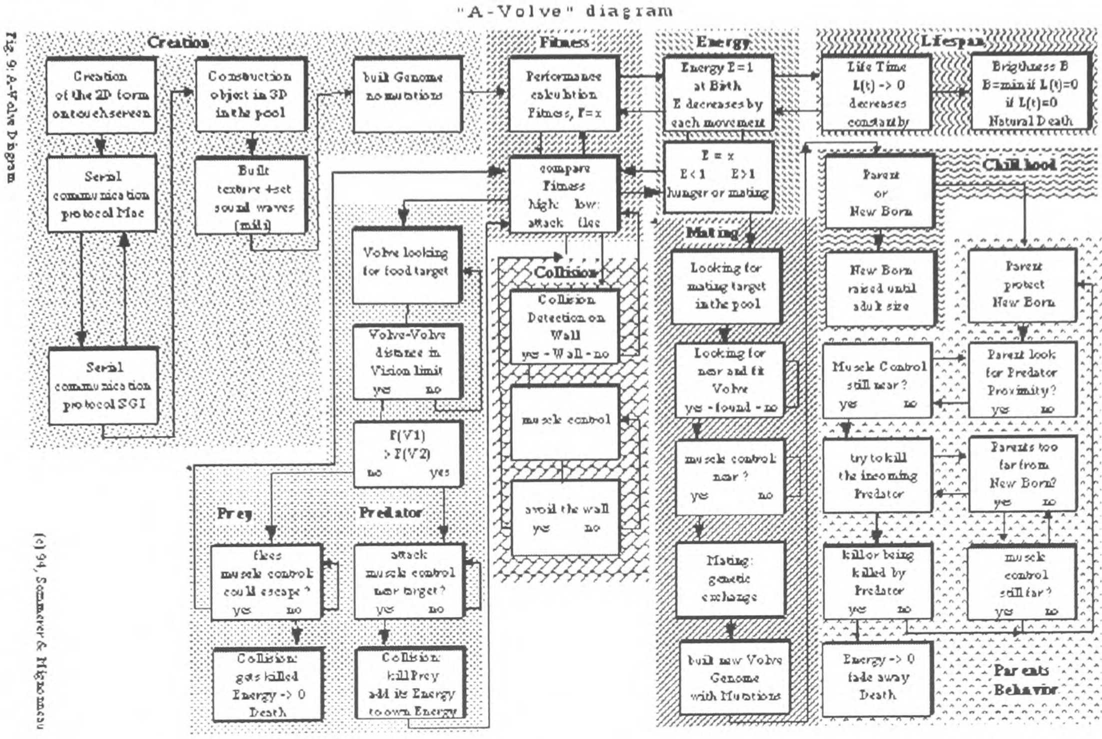


Fig. 25 Diagram of A-Volve's internal decision parameters.

Although the system's evolution could take place by itself without external influence, the system is designed to allow human interaction as well. A camera detection interface developed by Laurent Mignonneau can detect the users' hands when they reach into the water of the pool. When a user touches a creature inside the water (the creatures are projected through a mirror onto the projection surface inside the pool, as shown in Figure 26), this creature becomes irritated and wants to escape. On the other hand, once caught, it will calm down and stop swimming, becoming "invisible" to the other creatures. By catching the creatures with their hands, users can for example protect a prey from an attacking predator, or stop the predator creature to attack a prey. Creatures in "A-Volve" have a limited life time (around one minute), and if a user protects a creature too long, its lifetime might run out and the creature would die. Users of the system thus not only make creatures by drawing them on the touch screen but also help them in their interactions or interfere with their internal motivations. The overall set-up of the system is shown in Figure 26.

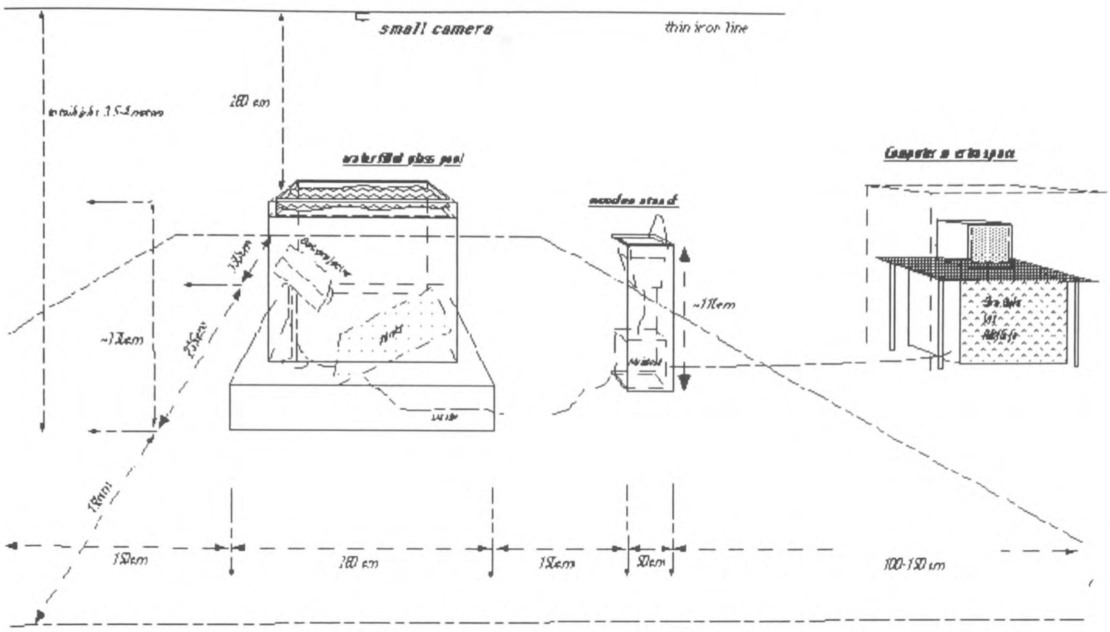


Fig. 26 A-Volve's system set-up showing touch-screen interface and water-filled glass pool with projection system that projects images of creatures into the pool.

To summarize, we can say that "A-Volve" was one of the first systems that allowed users to not only watch the interaction and evolution of artificial creatures (as demonstrated by Terzopoulos *et al* (1995), Karl Sims (1991, 1994) or Todd and Latham (1992)) but also physically interact with these creatures by touching them and getting involved in their internal evolutionary mechanisms. "A-Volve" was designed as a multi-user interactive experience for the general public and has been shown in numerous media museums and science museums around the world since 1994. A detailed description of "A-Volve" and its internal organization and algorithms is available in Sommerer and Mignonneau (1997a, 1997 b).

5.1.5 – TechnoSphere

In 1995, a group of British artists (Mark Hurry, Jane Prophet and Gordon Selly, 1999) launched an Internet site called "TechnoSphere", an on-line interactive system where users could assemble creatures from certain defined body parts (heads, bodies, legs, etc.) and then send these creatures to the "TechnoSphere" environment and watch them interact and evolve. In this system, users create their own artificial life forms, building carnivores or herbivores from component parts (heads, bodies, eyes, and wheels). Their digital DNA, or genetic specification, is linked to each component part, determining speed, visual perception, rate of digestion and so forth. Once a creature is built, users name their digital creature and it is tagged with their email address and put into the 3-D world. As the creatures grow, give birth, move, evolve and die they send brief email messages, postcards "home" to the users that designed them, describing the key events in their artificial lives. Users can visit the website and see 2-D snapshots of their beast at any time, check family trees and world statistics, and trace other creatures and the users that designed them. For example, users might be interested in finding out more about a creature which their beast had interacted with, so they can use the ID number of the other creature, which is sent in the email messages to track that creature down (Hurry *et al.*, 2000). Figure 27 shows a snapshot of the "TechnoSphere" world with several creatures inhabiting it.

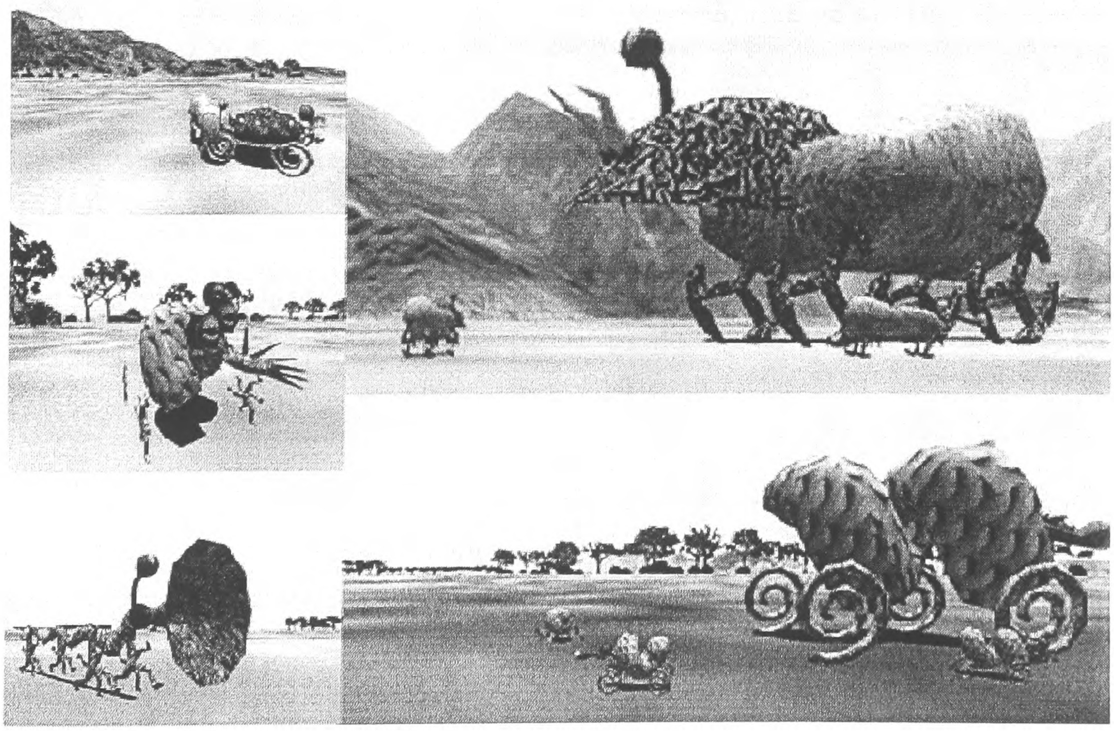


Fig. 27 Screenshots from "TechnoSphere" world.

5.1.6 - Ventrella's Darwin Pond

Another software program dealing more strictly with artificial evolution and user interaction possibilities was "Darwin Pond", developed by Jeffrey Ventrella in 1996. Artificial creatures, called "swimmers", perform tiny histories of the evolution of swimming in virtual waters by genetically "discovering" ways to get around (Ventrella, 1996, 1995). This system is based on using genetic algorithms (GA) to automate animated motion by evolving stimulus-response mechanisms that optimize the locomotion of physically based figures (see also: Ngo and Marks, 1993 and Sims, 1994). Figure 28 shows a screenshot of Darwin Pond and the graphical user interface that allows users to interactively control the evolution of the "swimmer" creatures.

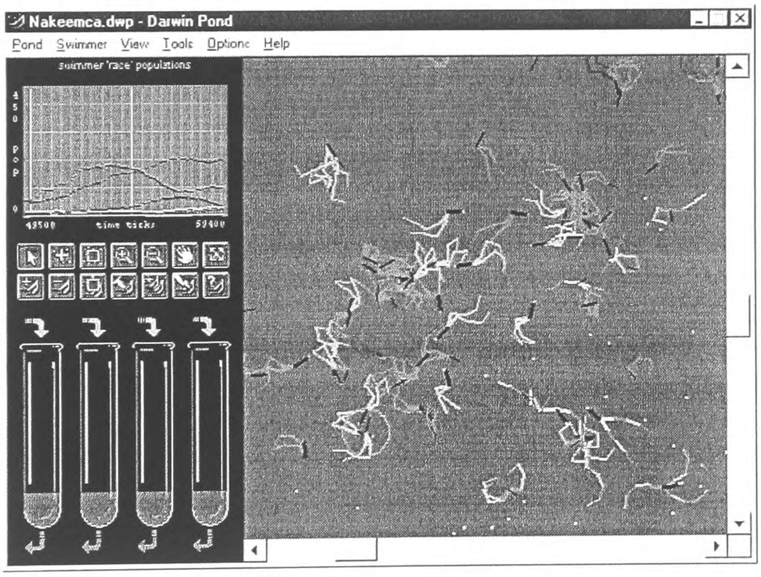


Fig. 28 Screenshot of Darwin Pond.

5.1.7 - Damer's Nerve Garden

Probably inspired by "TechnoSphere" (Section 5.1.5) and Prusinkiewicz's work on artificial plant algorithms (Prusinkiewicz *et al.*, 1990), Bruce Damer developed an on-line system called "Nerve Garden" in 1997 (Damer *et al.*, 1999). It is a biologically-inspired multi-user collaborative 3D virtual world, where users can operate a Java client, called the Germinator, to extrude 3D plant models generated from L-systems.

The 3D interface in the Java client provides an immediate 3D experience of various L-system plant and arthropod forms. Users employ a slider bar to extrude the models in real time and a mutator to randomize production rules in the L-systems and generate variants on the plant models. In this way, various plant extrusions can be produced by using the Germinator. After germinating several plants, the user selects one, names it, and inputs it into a conventional VRML 2.0 scenegraph called the Seeder Garden. Figure 29 shows the Nerve Garden web browser and a typical landscape scene.

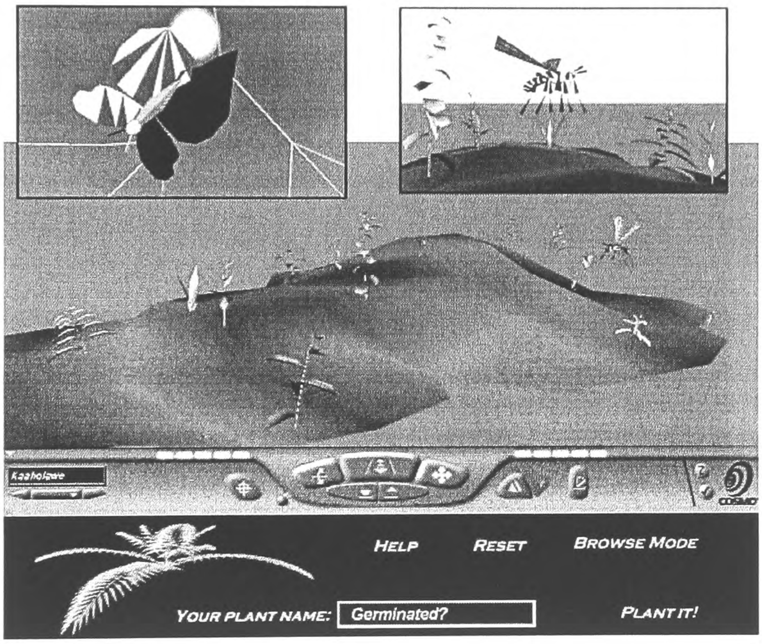


Fig. 29 "Nerve Garden" screen shot.

5.1.8 - NearLife's Virtual Fishtank

In an effort related to the "A-Volve" concept and interface design (Sommerer & Mignonneau, 1994), a group of MIT researchers formed the "NearLife" company and created a system called "Virtual Fishtank" (1998). The artificial fish-like creatures here are also created by users on a touch screen and also interact with each other in a virtual pool displayed on wall screens. In comparison to "A-Volve," the design of these virtual creatures is rather pre-defined, as users can only assemble parts of the creatures' bodies. Apparently, this system also draws on the work on artificial fish by

Terzopoulos. In his work, Terzopoulos *et al.* (1994) modeled realistic looking artificial life fishes that are able to locomote, sense, learn, school and prey, and interact with each other. Figure 30 shows a screenshot of the on-line version of "Virtual Fishtank."

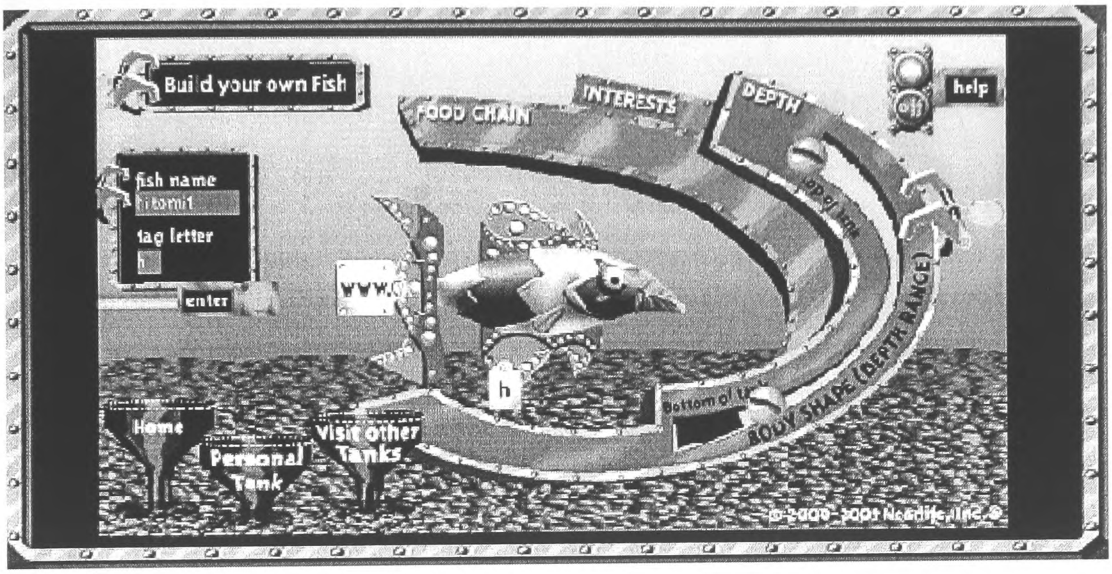


Fig. 30 "Virtual Fishtank" screen shot.

5.1.9 - McCormack's Turbulence

In 1994, John McCormack created an interactive installation called "Turbulence." "Turbulence" is an interactive video laser disc containing computer animations of synthesized forms designed by McCormack, where genetic algorithms were used to produce artificial life forms. Users can interact with this system by pressing words and symbols on a touch screen and thus triggering different selections from the videodisc. According to McCormack, "Turbulence" develops and examines abstractions of life-like processes, visualised as geometric entities manufactured from a deterministic set of digital instructions applied millions of times by a computer [...]. He says this approach makes it a poetic interpretation that draws upon the philosophical implications of evolutionary theory. As an interactive museum, it is a collection of abstract thoughts, simulations, ideas, information and poetry—all a multiplex of links into an interactive web of computer synthesized imagery

(McCormack, 1994). Figure 31 shows an example screenshot from the "Turbulence" interactive laser disk.

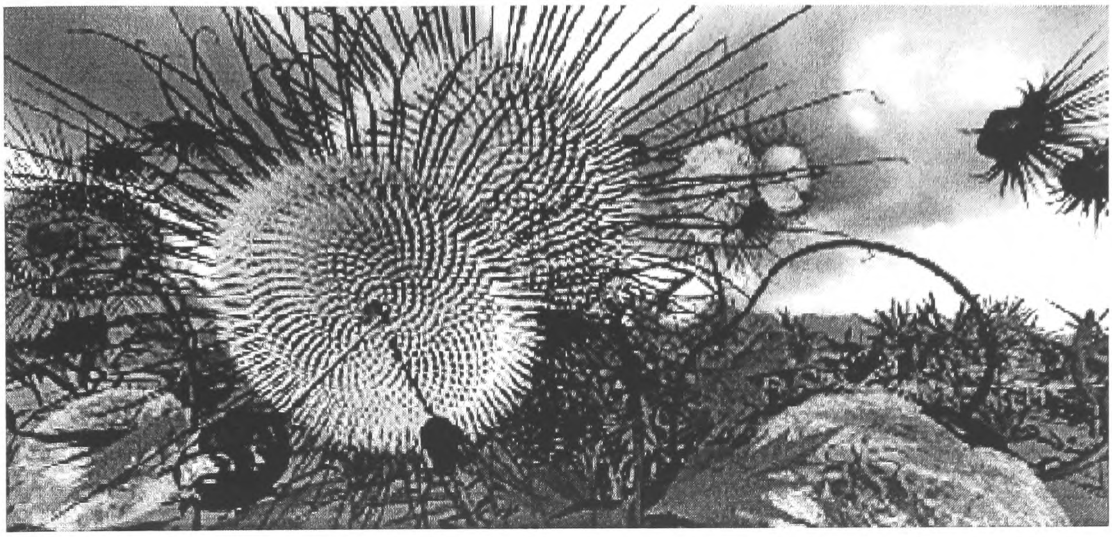


Fig. 31 Screenshot from McCormack's "Turbulence" laser disk.

5.1.10 - Innocent's Iconica

In 1998, Australian Artist Troy Innocent developed an interactive system called "Iconica" which, inspired by Visual Language Studies, constructs a virtual world based on information. In this system iconic elements are used as basic building blocks to create a unique iconic language. Any event or object in this world can then be described by the users, who may also recombine the elements using grammatical rules to create a wide range of possible meanings and expressions. Accordingly, the world has the capacity to evolve, change and mutate through human interaction and its own evolutionary process (Innocent, 1998). Figure 32 shows a screenshot of the language interface of "Iconica", including several of the iconic language symbols.

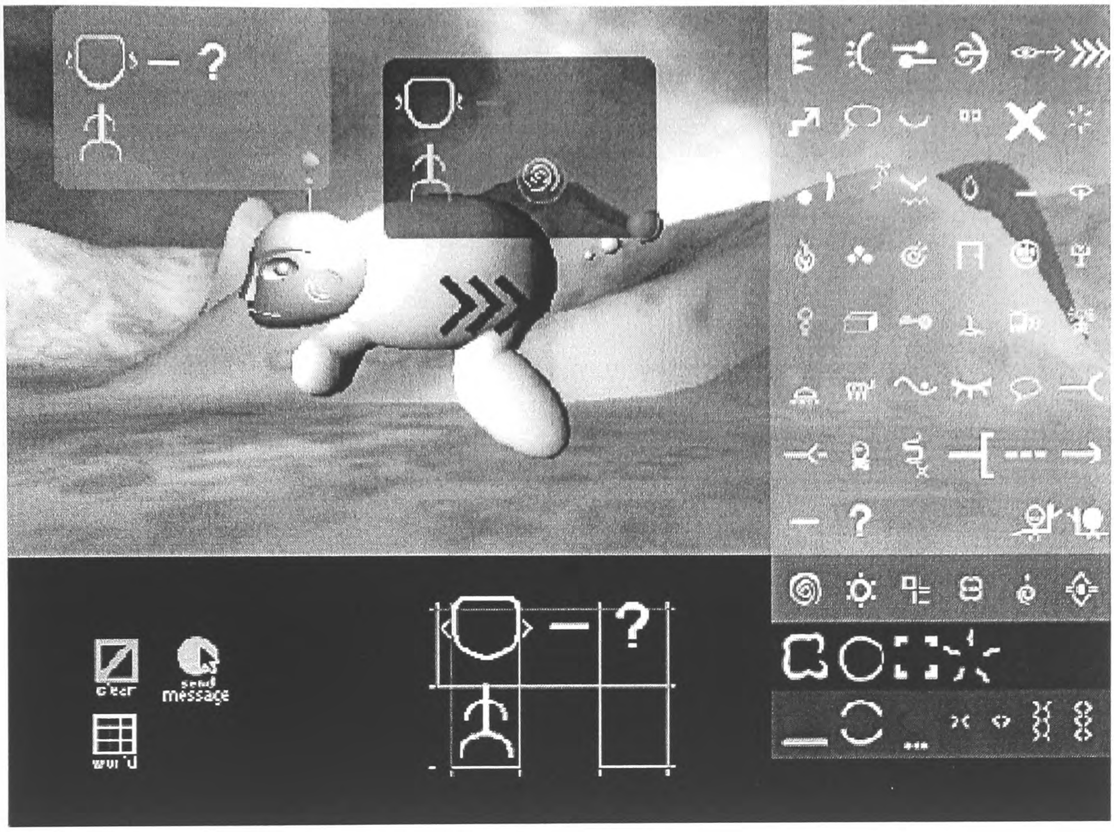


Fig. 32 Language Interface of "Iconica."

5.1.11 - Annunziato's Relazioni Emergenti

In 2000, Italian artists Mauro Annunziato and Pierre Pierucci created an interactive artificial life system called "Relazioni Emergenti" (Emerging Relations). Here, users are represented as artificial life graphical filaments that can interact with each other, exchange information, and reproduce (Annunziato *et al.*, 2000).

In this installation, a retro-projection screen displays the images and a video camera detects the user's position. The user's movement and motions are used to give "energy" to the single filaments, and they develop at the location that represents where the user is standing. Filaments created by other users remain visible, and each filament also contains a sound message. The observer can see filaments growing at the location where he or she is standing and can also interact with filaments that other observers have already created. A view of the installation is shown in Figure 33.

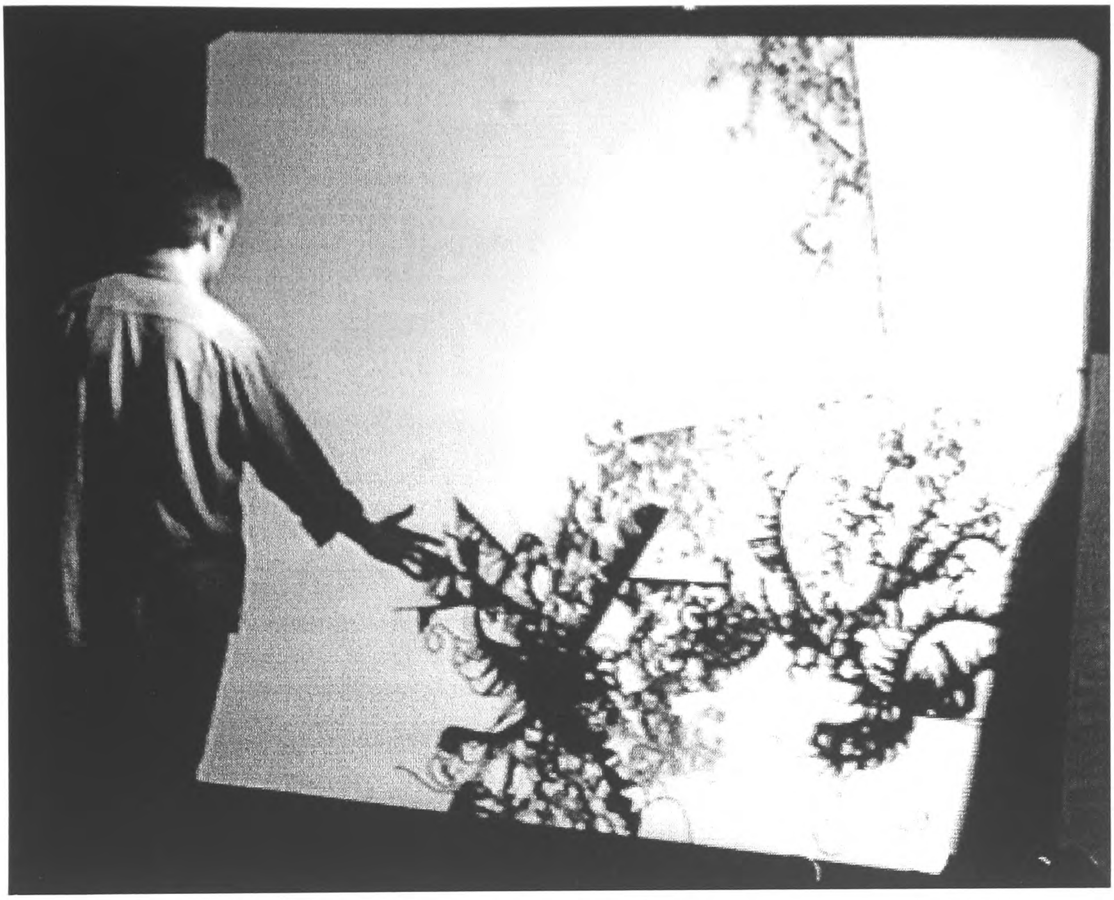


Fig. 33 View of "Relazioni Emergenti" installation. The user's position is linked to the development of graphical filaments.

Annunziato considers the developing graphical and acoustical patterns to be structurally complex, and through genetic mutations among the filaments graphical and sonic populations can evolve. While he calls these new patterns emergent properties, he also points out the possibility that these self-organizing properties and the emergence of new shapes and sounds are mostly perceived in the mind of the observer. A screenshot of such emergent patterns is shown in Figure 34. According to Annunziato, "the user's perception is able to reconstruct the hidden order without having a clear consciousness of it", and for him "the main objective of the artwork is to build a metaphor of the world of the communication webs and the mechanisms of formation of collective messages" (Annunziato *et al.*, 2000).

Fig. 34 “Relazioni Emergenti” installation screen shot with emerging patters of graphical filaments.

5.1.12 – Greenfield’s Co-evolutionary Image Approach

In 1992, parallel computing pioneer Daniel Hillis introduced the first time co-evolutionary techniques to solve optimization problems involving sorting (Hillis, 1992). Spurred on by this work, mathematician Gary R. Greenfield in 2000 modeled a system that could automatically evolve images from a random group of seed images by injecting them with parasite parameters. Based on the assumption that for humans “visually interesting images are the ones that cause our filtering apparatus—our eyes—to generate anomalies for our brain to process”, Greenfield devised a system that attaches simple digital filters to fixed locations on a seeding image, convolves local portions of the image with the filters, and then compares the convolved image with the original seeding image. Greenfield’s system searches for images where the convolved image is significantly different from the original. The filter is parasitic to the image by attempting to blend with it, while the host image attempts to repel the parasite by making it visible as a blemish (Greenfield, 2000).

Greenfield's seeding images are only 100 x 100 pixels, and during initialization he fixes a number of locations for parasites to attach to. He then generates a random host population of images and attaches a randomly generated parasite to each of the fixed locations on each host image. The parasite populations are then managed according to the location they are specific to. At each time step, fitness updates are calculated and the least fit hosts are removed from the population of images. Random matings between the survivors are used for replacements. Similarly, for each location, the least fit parasites are removed and their replacements are determined by cloning and mutating the fittest survivors from that location's population. A new host inherits the parasites that were attached to the host it is replacing. Since a host's parasites only act by filtering a small patch on the host, and since the phenotype does not have to be generated in visual form, the coevolution implementation is very fast. This coevolutionary approach interjects seeding images with parasites and then evolves "visually interesting" images, based on the assumption that more complicated or complex images should evolve. The results of this operation are shown in Figures 35 and 36.

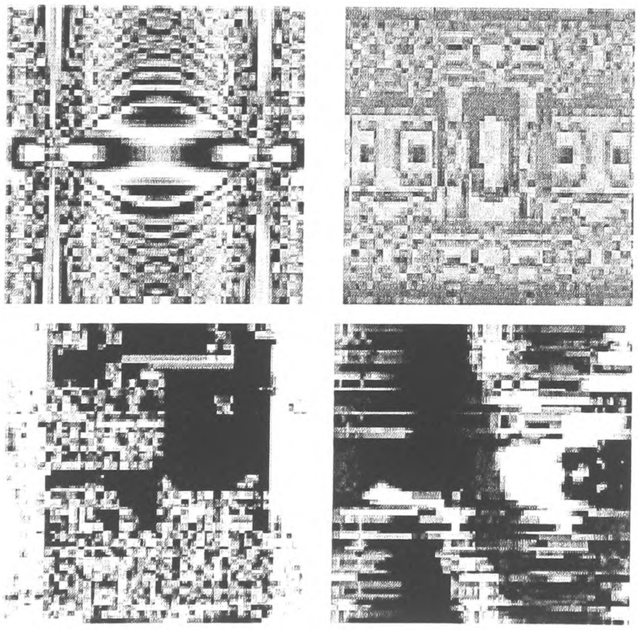


Fig. 35 Greenfield's Co-evolutionary Image Approach: the figure shows coevolved images from four different runs. All were evolved starting with small random populations. The system is able to produce diverse imagery by exploring different evolutionary trajectories in image space.

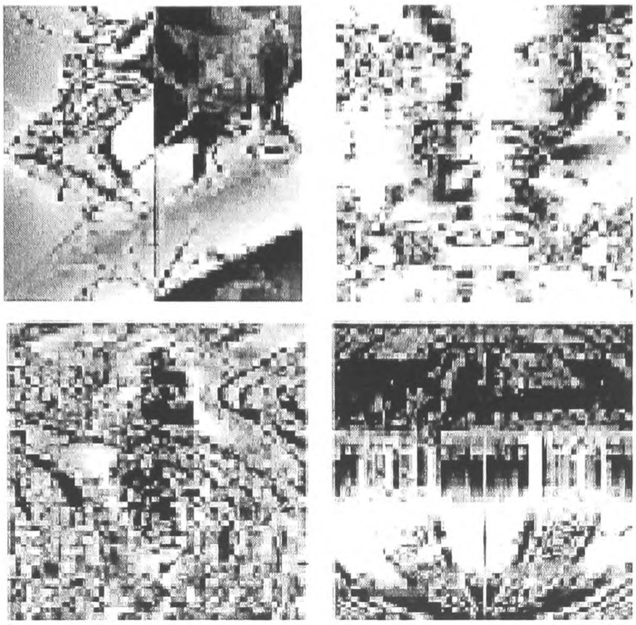


Fig. 36 Greenfield's Coevolved images from four runs using larger genome sizes but with the simulation run for only 1500 generations. Host population size is thirty with three parasites per host. The fittest images are culled for inspection every 200 generations.

5.1.13 - Other Artificial Life Artworks and Demonstrations

It would exceed the scope of this thesis to mention all of the artists and designers who subsequently became interested in applying artificial life techniques or metaphors to their artworks. A list of artists' names and projects is provided at Telefonica Spain's (1999, 2000, 2001) "Vida/Life 2.0 and 3.0" competition home page, and more information is also available at the Biota home page (Biota, 2001) and the International Generative Art and Design Conference home page (2001). A good overview and collection of the ever-expanding list of artists and programmers who use artificial life techniques for image creation is also provided at Craig Reynolds (2001) home page, and the writings of Mitchell Whitelaw (2000) provide a good introduction to this field.

A few artists and scientists and their projects shall be mentioned here. In 1998, fractal image specialist Ken Musgrave created a project called "Genetic Programming, Genetic Art: Dr. Mutatis", which provides Karl Sims-like evolution of images by using esthetic selection (Musgrave, 1998). Steven Rook is another pioneer artist who already in the early 90s used evolutionary art and aesthetic evolution of algorithmic images (Rook, 1998 & 2000). "Genetic Art" by Peter Kleiweg (1998) is also based on Karl Sim's genetic algorithms but uses PostScript language instead of LISP language. John Mount's "Genetic Art III" project finally virtualises Sims' Genetic Images installation (see Section 5.1.2), allowing web users to act collectively as aesthetic selectors by evaluating the images displayed (Mount, 1998). Furthermore, at Mattias Fagerlund's website users can evolve their own images (Fagerlund, 2001).

Yoshiaki Ishihama (Stone) uses genetic programming for the interactive evolution of Genetic Fractals (Ishihama, 2001). Other research-related works have been done by Duncan Rowland, who controls the appearance of 3D computer graphics objects via genetic algorithms to investigate aesthetic preference for 3D facial surfaces (Rowland, 1998). Riccardo Poli and Stefano Cagnoni use genetic programming for image enhancement (Poli *et al.*, 1997).

Other A-Life inspired art works include Nik Gaffney's "Mutagen" project, a form-breeder which allows both user-driven and autonomous evolution (Gaffney, 1998), Andrew Rowbottom's "FORM" software (Rowbottom, 1998), and the "Cybertation and Dancer DNA" project by The Zen Room group (The Zen Room, 1998).

In 1998, Jason Spofford released his "Primordial Life" software, an experiment in computer-based evolution where users can evolve artificial life forms called biots that struggle against each other in a battle of survival (Spofford, 1998). In 2000, a French scientist (Heudin, 1998) developed a system called "Life Drop", where users can interact with simple 2D creatures over the Internet and watch them evolve (Life Drop, 2000).

5.2 - Overview of Generative Musical Systems

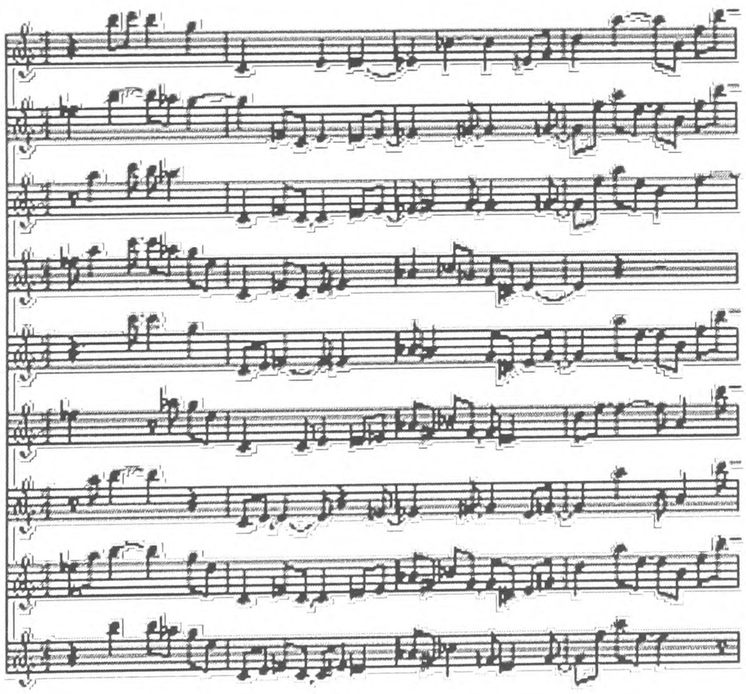
Generative processes and Artificial Life techniques have been applied to music as well, and Alistair Riddell (Riddell, 1999) notes that "Applied process is becoming a kind of defacto theory of contemporary art because it can suggest and define a set of possibilities for artistic production irrespective of whether sound is manifest by any artist or not. With interest in music as data, it appears that the creative design of musical processes might become an art in itself". It would exceed the scope this thesis to mention all composers and musicians who have applied Genetic Algorithms (GAs), Genetic Programming (GP), or Generative Processes to musical composition, but a few prominent composers/musicians shall be briefly mentioned here.

5.2.1 - Nelson's Sonomorphs

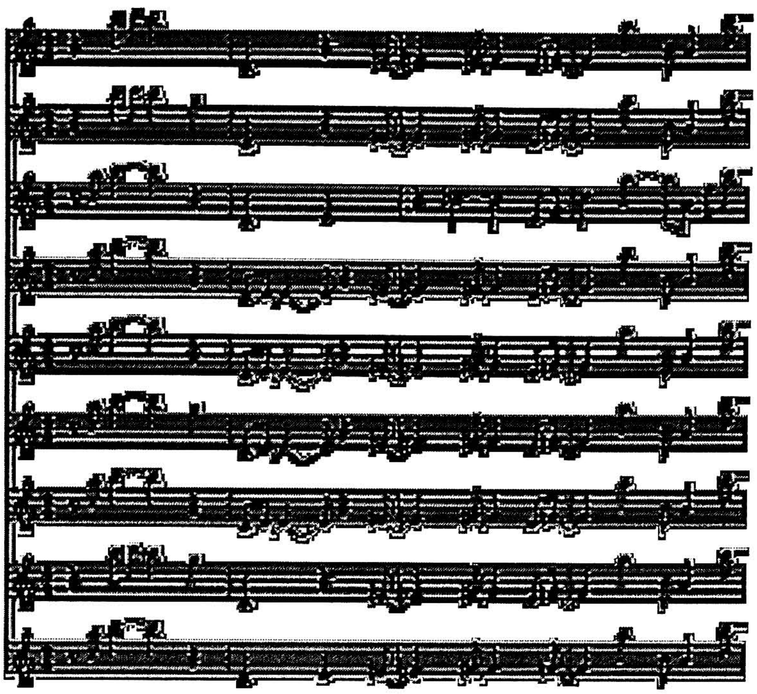
As early as 1992, Gary Lee Nelson applied genetic algorithms to the growth and development of musical organisms (Nelson, 1992). "Sonomorph" describes elements of musical compositions whose size ranges between a motive and a phrase. Nelson uses a variety of strategies for building the genetic code and selection criteria, and as a result he has developed a body of pieces where musical phenotypes evolve through a combination of reproduction and genetic mutation. The genetic code that is passed to each generation is interpreted with an algorithm that generates musical structures. This genetic code is mapped onto musical parameters and presented to the composer for subjective aural evaluation. The sonomorphs that "survive" become the progenitors of future populations.

The population in this experiment consists of an array of individuals arranged on a square grid of variable size. The genome for each individual is a one-dimensional array of 32-bit integers. The length of the genome is variable. The population is initialized by filling all of the genomes with pseudo-random numbers. A bit string is used to define a rhythmic pattern. A one (1) means play a note; a zero (0) means don't play. The grid is divided into variable-sized "islands". The initial state of the "world" is set by global controls. Grid dimension and genome size are needed to generate the initial random population. Island size and walk length govern the children's field of search for parents. Probabilities for migration, crossover, and mutation characterize and influence evolution through the generations. Nelson's experiments were carried out with the MAX.5 software, an objected-oriented language for programming interactive musical processes.

To select suitable parents, empty-gene children search for potential parents by performing a random walk around their island. If successful, two parents are selected and their genomes are subjected to the crossover algorithm and the resulting child is passed through the mutation function, as already described in Chapter 4. The results of this evolutionary process can be seen in Figures 37 a and b, a) showing the original random initial population and b) showing the same island or score after 30 generations. One effect of the evolutionary process was the fitness selection for fewer bits, and the number of notes decreased drastically as well (Nelson, 1992). Figure 37 shows the evolution of sound through the Sonomorphs system.



(a)



(b)

Fig. 37 Sound evolution through the Sonomorphs system.

5.2.2 - Miranda's CAMUS Cellular Automata MUSic Generator

In 1990 composer Eduardo Miranda created CAMUS, a cellular automata-based music generator, using the Game of Life (invented by John Horton Conway) and the Demon Cyclic Space (designed by David Griffeath) algorithms. These shall be described here briefly.

5.2.2.1 - Conway's Game of Life

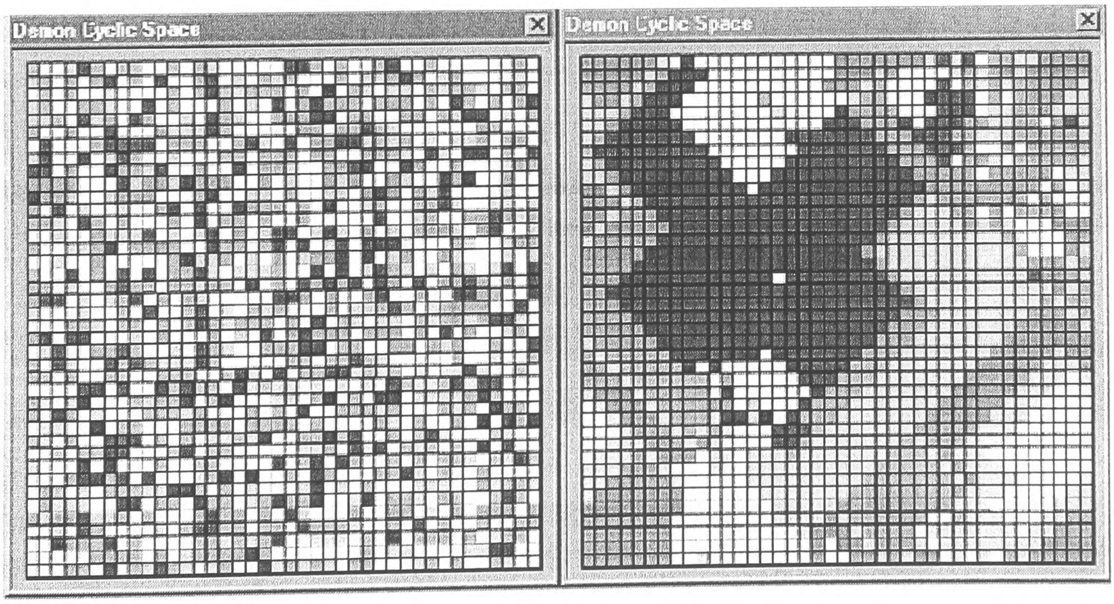
From one tick of the clock to the next, the cells of the Game of Life cellular automaton can be either alive (i.e., black) or dead (i.e., white), according to the following rules devised by Conway, as described in Gardner (1970):

> If a cell is dead at time t, it comes alive at time t+1, if it has exactly 3 neighbours alive.
>
> If a cell is alive at time t, it dies at time t+1, if it has fewer than 2 or more than 3 neighbours alive.

These rules are applied simultaneously to all cells of the lattice. An initial configuration of live cells may either grow interminably, fall into cyclic patterns, or eventually die off.

5.2.2.2 - Demon Cyclic Space

The rules of the Demon Cyclic Space cellular automaton, devised by David Griffeath *et al.* (1991), generate miniature worlds of incredible complexity. Initialised as a random distribution of coloured cells, as shown in Figure 38 a, it always ends up with stable, angular spirals reminiscent of crystalline growths, as shown in Figure 38 b. Each of the n possible states for a cell is represented by a different colour and numbered from 0 to n-1. A cell that happens to be in state k at one tick of the clock dominates any adjacent cells that are in state k-1, meaning that these adjacent cells change from k-1 to k. This rule resembles a natural chain in which a cell in state 2 can dominate a cell in state 1 even if the later is dominating a cell in state 0. However, since the automaton is cyclic, the chain has no end and a cell in state 0 dominates its neighbouring cells that are in state n-1.



(a) (b)

Fig. 38 Examples of a Demon Cyclic Space cellular automaton.

5.2.2.3 - CAMUS Cellular Automata MUSic Generator

CAMUS uses a Cartesian model to represent a triple, a set of three notes. The model has two dimensions, where the horizontal coordinate represents the first interval of the triple and the vertical coordinate represents its second interval. The system uses both automata in parallel to produce music. The Game of Life automaton produces triples, and the Demon Cyclic Space automaton determines the "orchestration" of the composition. In this case, each colour corresponds to an instrument (MIDI) designated to perform the notes generated by a specific cell. Each musical cell has its own timing, but the notes within a cell can assume different durations and can be triggered at different times. An example of the CAMUS architecture is shown in Figure 39. To begin the music process, the Game of Life automaton is set up with a starting configuration, while the Demon Cyclic Space automaton is intialised with random states, and both are set to run. At each time step, the coordinates of each live cell are analysed and used to determine the triple which will be played at the corresponding time in the composition. The state of the corresponding cell of the Demon Cyclic Space automaton is used to determine the orchestration of the piece (Miranda, 200).

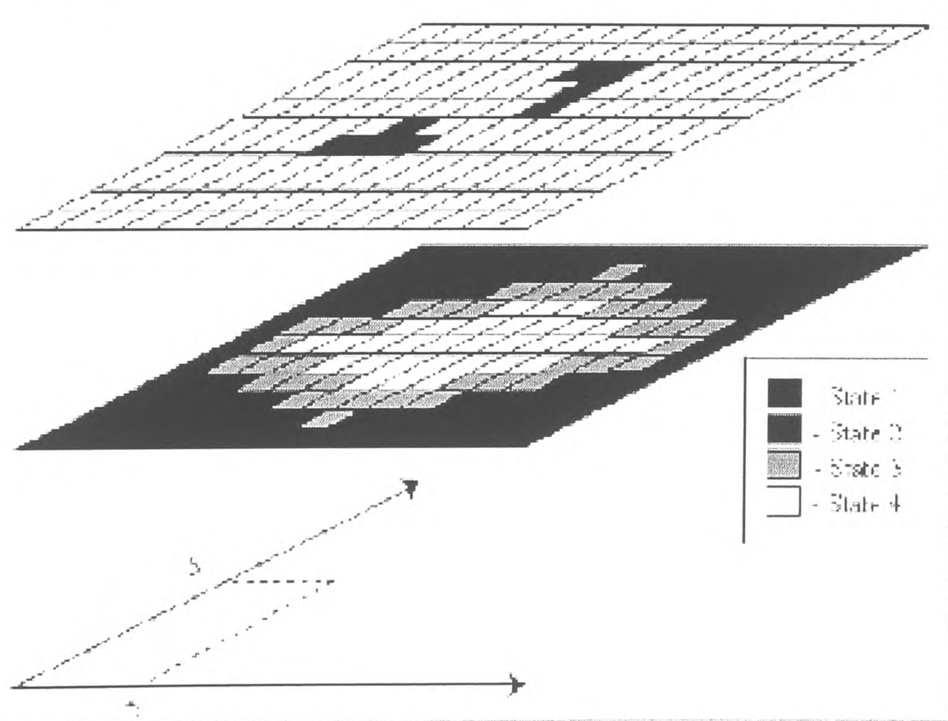


Fig. 39 CAMUS cellular automata architecture.

5.2.3 - Bilotta's Synthetic Harmonies

Bilotta *et al.* (2000) note that many musical systems have been realized using techniques related to the mathematical models of Artificial Life and that it is possible to organize the following musical systems' taxonomy:

Fractal music
Music produced by chaotic structures: Chua's circuit
L-systems: Used to generate Midi files
Generative music based on grammars
Evolutionary music based on genetic algorithms
Genetic music: using DNA sequences to generate MIDI music
Music generated by Cellular Automata

Bilotta's own system, called "Synthetic Harmonies" is also based on the idea of linking cellular automata with genetic algorithms to produce sounds that can subsequently be evaluated by groups of listeners and then reproduced according to their musical fitness.

The system has a dimension of 100, where the CAs have 8 states and both the initial state and the transition rules are generated randomly. Each column of the cellular automata pattern corresponds to a specific note. The Cell State represents the starting offset for that note with respect to the beginning of the bar fraction that the current row represents. The notes belonging to a row are all played in the time fraction assigned to the row itself. The other parameters, such as tempo and instrument assignments, are chosen in the MIDI rendering window. The parameters the user can change are the number of cellular automata state transitions per musical bar, the musical tempo model, the number of beats per minute, and the instruments split definition and assignment.

This system can generate five MIDI files by using two random initial states, so ten families of five MIDI files each are obtained. The listeners then judge the pleasantness of the sounds in each family. A family's fitness can be derived by summing the scores obtained by each file in the family, and the families that obtain the greatest fitness are allowed to evolve. This process is then repeated for ten generations. As a result, the music generated is varied but not very significantly. Bilotta *et al.* [2000] note that the evolutionary process might have to be repeated a higher number of generations for natural selection to be effective and to overcome the effects related to the initial state and the rendering process. As an overall result, "Synthetic Harmonies" tries to define and explain "musical fitness", which is generally operated more intuitively by the listeners; this value is apparently also based on the theory of Pythagorean consonance, which demonstrates that there is a biological basis for consonance (Zentner *et al*, 1996).

5.2.4 - Dahlstedt and Nordahl's Living Melodies

Using not cellular automata (CAs) but genetic algorithms (GAs) to create music, "Living Melodies" was developed by Swedish composer Palle Dahlstedt and artificial life researcher Mats Nordahl (Dahlstedt & Nordahl, 1999) as an artificial world containing co-evolving and communicating creatures that use the sounds produced by the entire system as musical material. In this particular world, creatures need to make sounds to find mates. The genomes of the creatures consist of a sound genome and a procedural genome. Both are allowed to vary in length during the evolutionary process. The world of the living melodies is a square lattice where the creatures occupy a single lattice site; only one creature can be located at a site. They can walk around, one step at a time, in eight different directions.

A lattice site can store food, and if a creature visits the site, it eats the food and thus increases its life points. A simple algorithm for sound propagation was used. The air is implemented as a two-dimensional array, which contains the current sound identity (note/interval), amplitude, and direction for every location. If a stronger sound appears, it overwrites the previous sound at the site. The sounds decay linearly with time. A number of criteria have to be fulfilled for two creatures to be allowed to mate. Creatures need to be located in neighbouring cells to mate, and they both need to have energy exceeding a specified lower limit and to be over a certain age. Moreover, the creatures are required to exercise occasionally and be happy to mate; in other words, they have to have heard some music they like recently, which ensures that sound production does not die out completely. As usual in genetic algorithms, the procedural genomes of the parents are combined by a crossover operation, which allows the genomes of the parents to be of unequal length. The number of genes in the offspring then ranges from the length of the shortest parent genome to the sum of the lengths of the two parents' genomes.

The results of the sessions of "Living Melodies" vary significantly, but as in many other evolutionary systems, the evolutionary pathways are not totally random, and some recurring patterns are observed. The artificial creatures generate interesting and musically useful sonic structures, and the system provides a good way to actually listen to the evolutionary process of simple creatures working together, triggering each other, and playing a small part each (Dahlstedt & Nordahl, 1999). Figure 40 shows a screenshot of the "Living Melodies" world and a musical result in piano roll notation, as seen on screen.

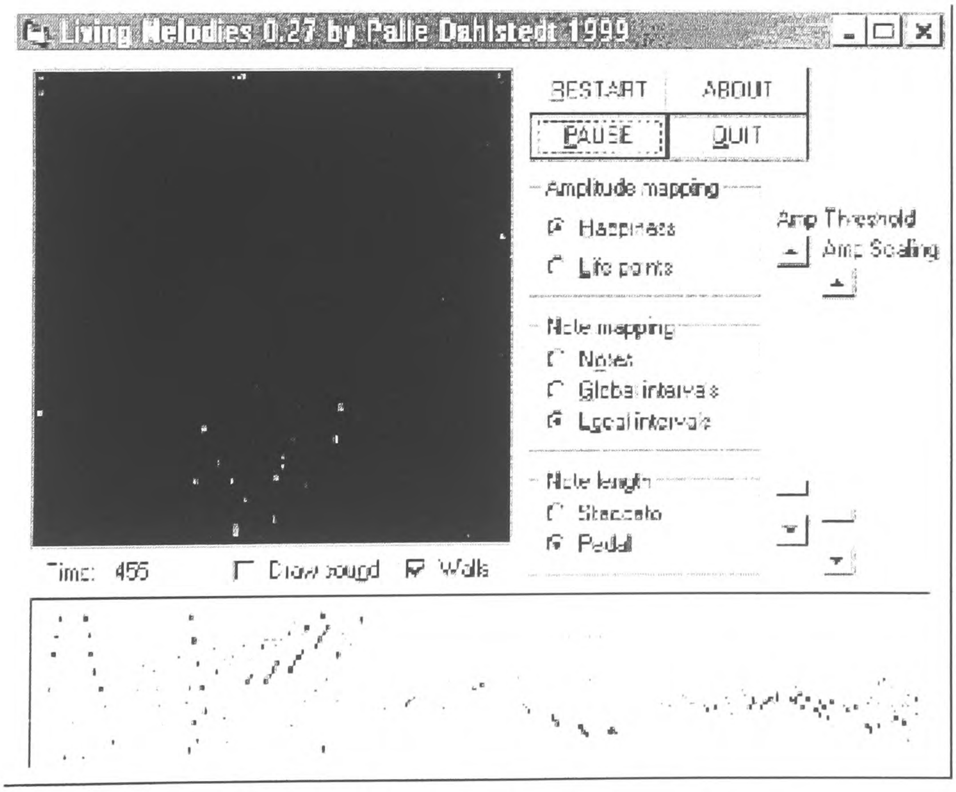


Fig. 40 Screenshot of "Living Melodies" graphical user interface.

5.2.5 - Other Generative Musical Systems

Lee Spector's "GenBebop" system uses genetic algorithms to produce interactive jazz music by applying a neural network fitness function (Spector & Alpern 1994). In a similar vein, the "GenJam" (Genetic Jammer) system developed by Al Biles is an interactive genetic algorithm that learns to play jazz solos. "GenJam" is a genetic algorithm-based model of a novice jazz musician learning to improvise. "GenJam" maintains hierarchically related populations of melodic ideas that are mapped to specific notes through scales suggested by the chord progression being played. As "GenJam" plays its solos over the accompaniment of a standard rhythm section, a human mentor gives real-time feedback to help derive fitness values for the individual measures and phrases. "GenJam" then applies various genetic operators to the populations to breed improved generations of ideas (Biles, 1994).

Another system in this category is Jason H. Moore's "GAMusic 1.0", a user-friendly interactive demonstration of simple genetic algorithms. Here, GAs generate short melodies and the user assigns their fitness. The iterative stepping, mutation frequency, and recombination frequency are all controlled by the user. Each series of musical notes is represented in binary form in an array of 128 elements in length. This allows a maximum 30 notes per melody and provides a solution space with approximately 3.4 * 10^38 possible melodies. The GA used by Moore is based on the simple genetic algorithm described by D.E. Goldberg (1989).

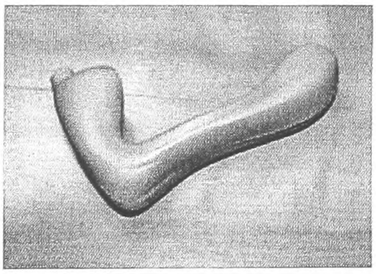
Other musical compositions that apply GAs are Alain Dorin's "Liquip-rism", a virtual prism of nodes which initiate musical events as they interact (Dorin, 1999), and computer graphic artist John McCormack's Grammar Based Music Composition (McCormack, 1996). More information on generative music and artificial life music can also be found in the "First Iteration Conference on Generative Processes in the Electronic Arts Conference Proceedings" CD-Rom, edited by Alain Dorin and John McCormack (Dorin & McCormack, 1999).

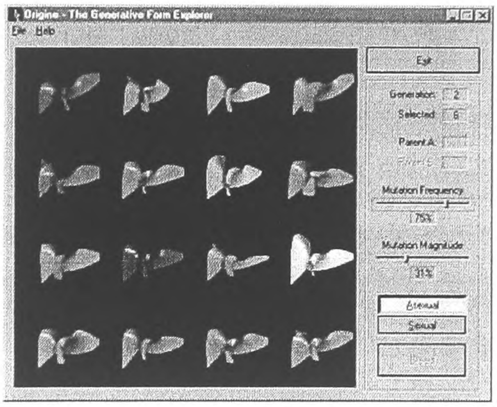
5.3 - Overview of Generative Design & Architecture

Artificial Life techniques have also recently been applied to product design and architecture. New so-called 'smart' products are emerging which combine familiar forms/functions with more flexible and adaptive control and inter-product communication capability. Agile and automated serial manufacturing processes and technologies can now economically fabricate high-quality products with a high degree of variation in the individual product instance. Increasingly sophisticated consumers are also demanding products which address individual tastes and lifestyle choices. The discovery and development of new models for consumer-centered product design will center on the results of applying interactive generative design techniques to product design and development. This approach to design redefines the consumer's role in the design and development process and holds the possibility for detailed and structured feedback to the designer and a more rapid and subtle product refinement than traditional marketplace and consumer research techniques (Pontecorvo, 2000).

Cristiano Ceccato (2000) notes that the concept of Parametric Design is concerned with generating design sets that exists within the boundaries of pre-set parametric values. Evolutionary Design, on the other hand, can help to extend the notion of parametric control by using rule-based generative algorithms to evolve common families of individual design solutions. These can be optimized according to particular criteria or used to form a wide variety of hierarchically related design solutions while supporting design intuition. The integration of Evolutionary Design with CAD-CAM, in particular for flexible manufacturing and mass-customization, provides a unique situation which exploits the full power of both approaches to create a new design-process paradigm. This new paradigm can generate limitless possibilities in a non-deterministic manner within a variable search-space of possible solutions.

As an example of this approach, Ceccato (2000) uses the Generative Design method to generate families of door handles. An Evolutionary Design tool uses a Genetic Algorithm (GA) to extract information that makes up a successful design by breeding families of related forms and testing them against a selective environment. The GA generates a new population of form data by generating a child population using breeding, cross-over, and mutation of the initial data. The generated data is translated into tangible form through the use of CNC or rapid prototyping machinery (Figure 41 a). The manufactured door handles are then tested in a real environment, in this case, an evaluation by users. Each of these users is required to give a verdict on various tactile aspects of each handle. These values are then calculated to a 'score value' and used as a fitness value for the GA in the next generation. A repetition of this sequence generates a collection of door handles, which reflects the user group's preferences within the design (Figure 41 b). This cycle creates a constantly changing family of related design solutions, which can either constantly adapt to changing criteria or converge on an emergent ideal configuration (Ceccato, 2000).



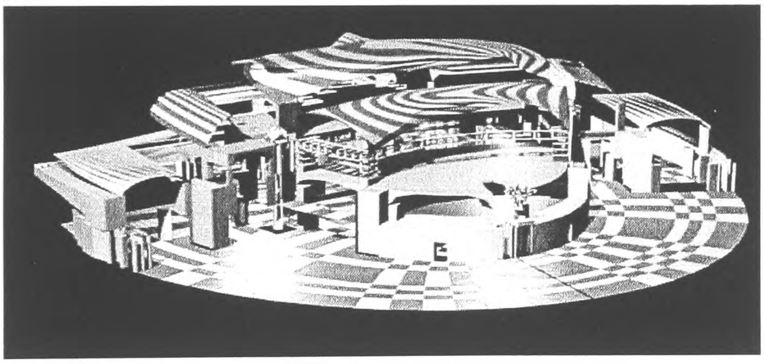


(a) (b)

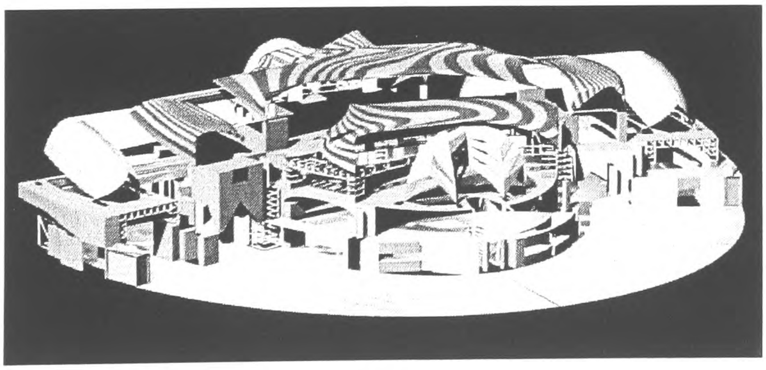
Fig. 41 Example of a GA applied to door handle design is shown in (b), and the resulting prototype produced through rapid prototyping machinery is shown in (a).

Architect and designer Celestino Soddu (2000) was instrumental in coining and propagating the term Generative Design and Architecture by organising the International Conferences on Generative Art in Milan, Italy (Generative Art, 1998-2001). He has applied artificial life techniques to the production of architectural spaces. Starting by reconstructing the architectural space of the Pantheon (built in Rome around 118 to 126 a.c., and shown in Figure 42 a) on a computer, Soddu uses the generative design approach to modify and mutate parts of the Pantheon's architectural structures, thus creating a whole range of architectural interpretations, as shown in Figures 42 b)-d).

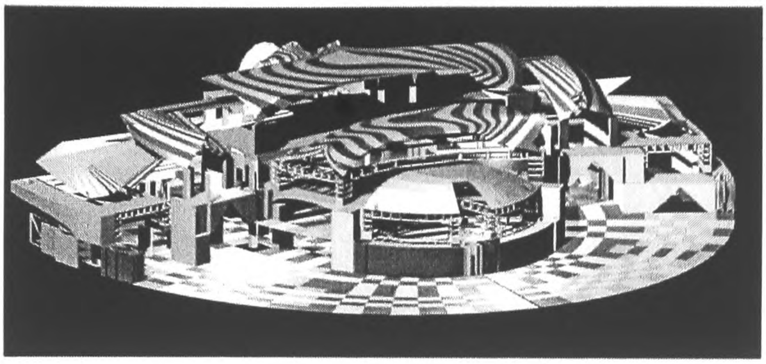
(a)



(b)



(c)



(d)

Fig. 42 (a) Shows an image of the Pantheon in Rome, Italy. Figures 42 (b)-(d) show three different interpretations of the Pantheon, generated through application of GAs.

5.4 - Overview of Generative Art Game Products

One of the most commercially successful application areas of Artificial Life is the gaming industry. Since the early 90s, game designers have recognised the entertainment potential of life-like systems for computer games, Internet- and telephone-based games, as well as entertainment, edutainment, infotainment and robotics. An ever-expanding number of games and software releases use Artificial Life and AI techniques and paradigms to create creatures, virtual characters, avatars and virtual worlds. For an introduction to this field, we refer the reader to an article on Games and Artificial Life by Lafarge (2000) and the writings on Japanese digital pets by Kusahara (2000). A few selected artificial life games, with focus on artificial evolution, shall be mentioned here briefly.

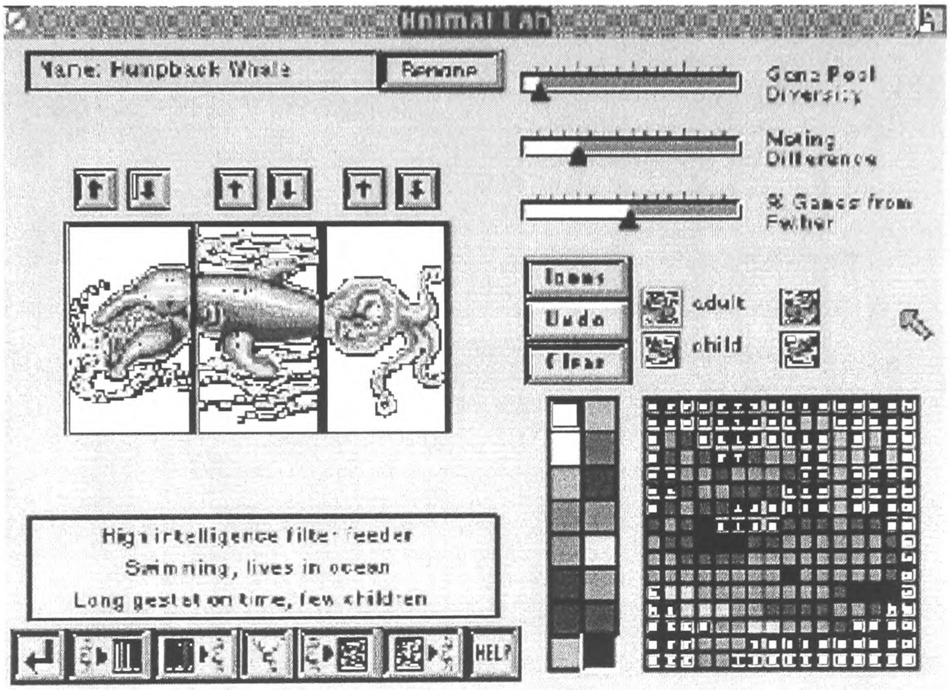
5.4.1 - SimLife

The first commercial software that exploited the ideas of artificial life was "SimLife", released in 1992 by Maxis software (SimLife, 1992). In this game, players explore the interactions of life forms and environments and manipulate the genetics of plants and animals. Players can also create new worlds with distinctive environments and see how certain species survive in these simulated ecosystems. Moreover, "SimLife" users can:

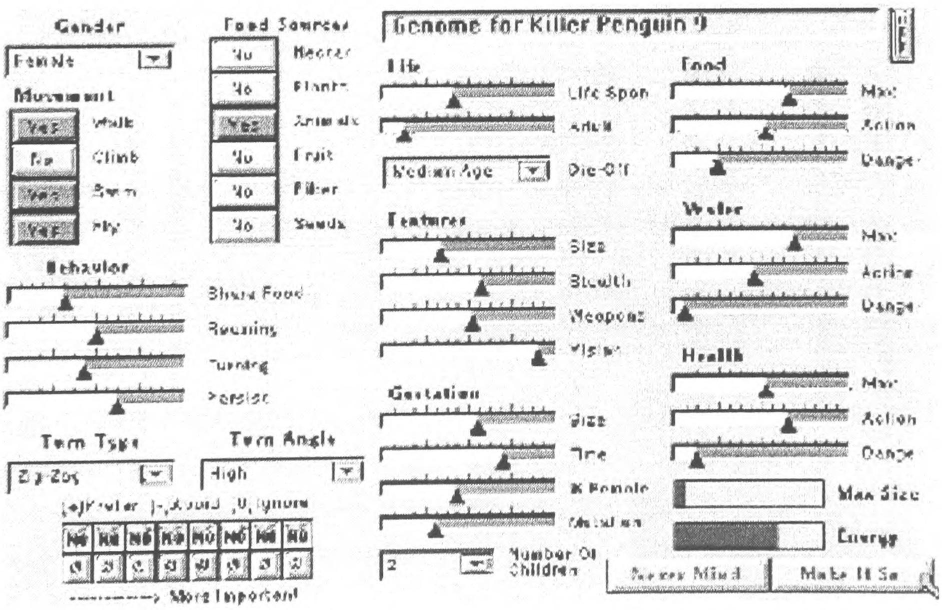
create and modify worlds
create and modify plants and animals at the genetic level
design environments and ecosystems
study genetics in action
simulate and control evolution
change the physics of the universe in the computer.

By manipulating the parameters in the various user interface panels, such as the "Animal Lab" panel (Figure 43 a) and the "Genome" panel (Figure 43 b), players can see the effects of mutation and interaction changes on the system's genetic pool

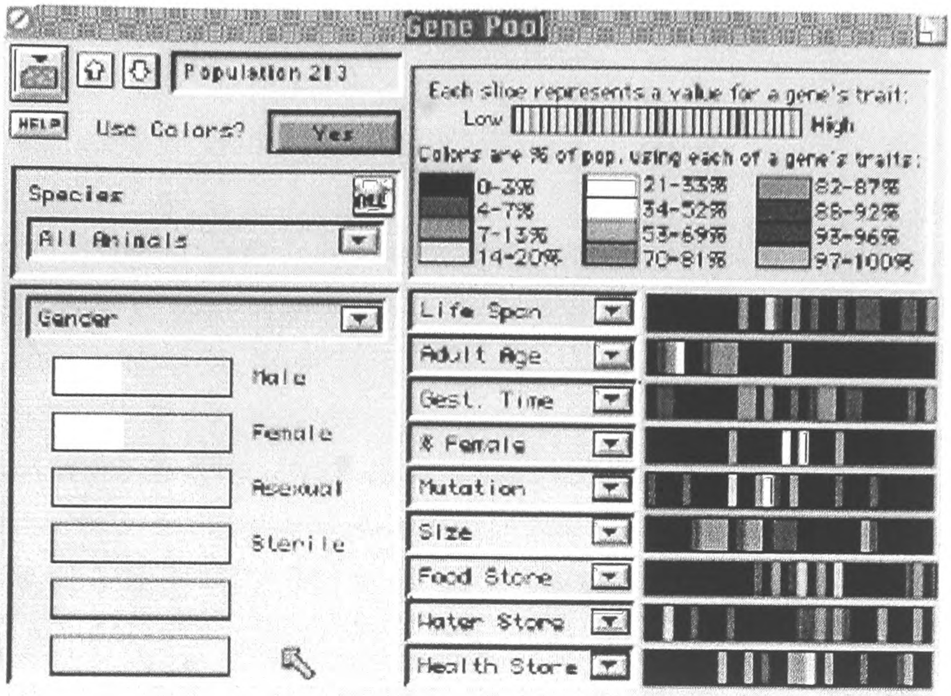
(Figure 43 c). Food webs and the effects of mutation, extinctions and natural disasters can be studied as well, all while playing with the various parameters of the gene pool, the ecosystem, and virtual life itself.



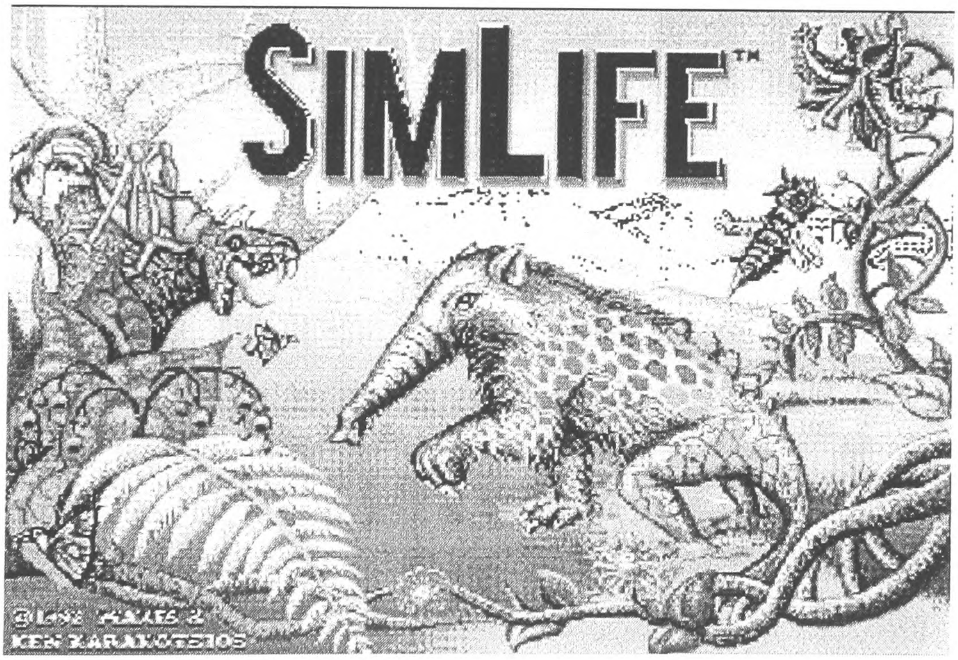
(a)



(b)



(c)

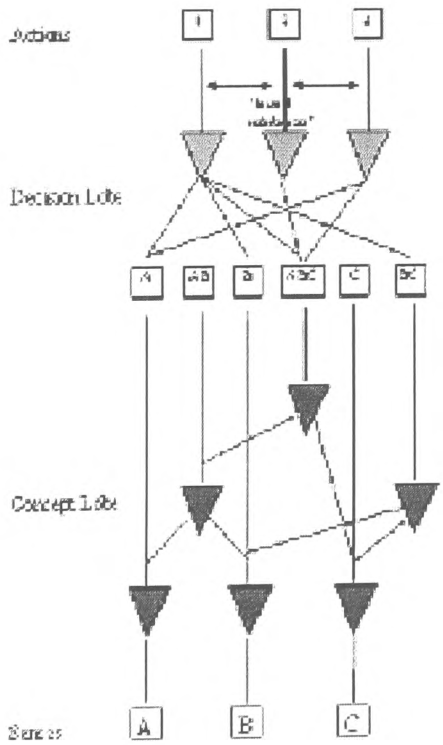


(d)

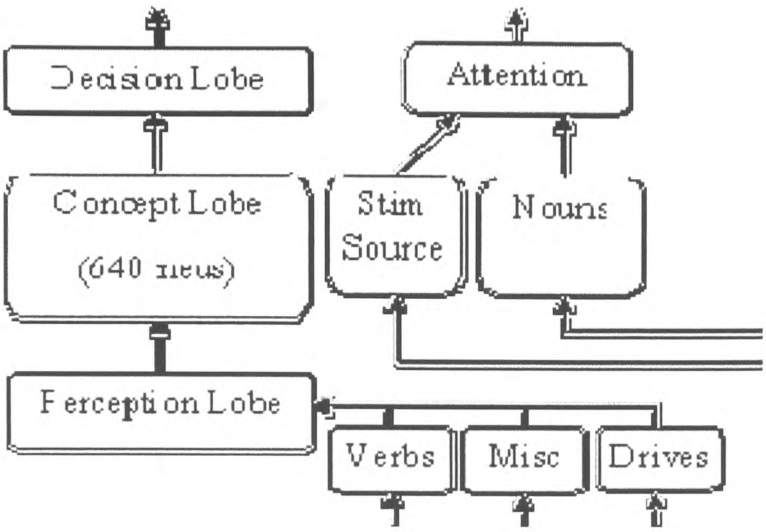
Fig. 43 SimLife's a) "Animal Lab" panel, b) "Genome" panel, c) "Genetic Pool" panel, and d) software package's title screen.

5.4.2 - Creatures

In 1996, a group of British researchers developed the commercial software called "Creatures" (Grand *et al,* 1997). "Creatures" is a simulated world inhabited by a number of synthetic agents, or creatures, that a user can interact with in real time. A creature's brain consists of several hundred neurons, clustered together in brain lobes which are connected by dendrites. Each lobe is a grid of squares, where each square on the grid is called a "neurone". Neurones are basically placeholders: they store information that the game engine then interprets. Dendrites link neurones in different brain lobes together, carrying information between the neurones. They travel one way only, from one neurone to another, so that a neurone can affect the state of another one but not vice versa. A creature's brain itself can be divided into three basic parts: information collection, information processing, and output, and the three corresponding lobes are the perception lobe, the concept lobe, and the decision lobe. Figure 44 illustrates the above neural network model to show this correspondence.



(a)

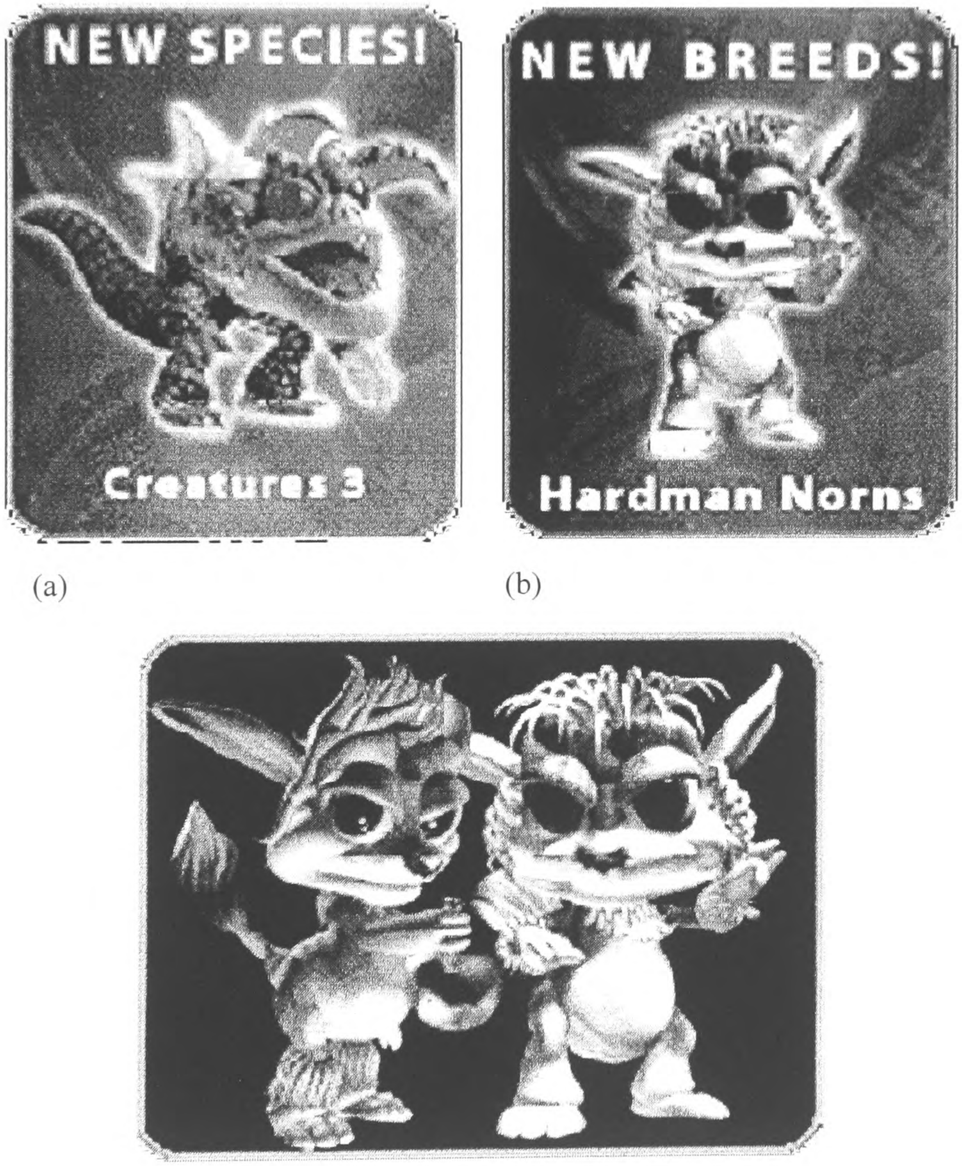


(b)

Fig. 44 "Creatures" neural network model (a), which controls a creatures' sensory-motor coordination and behavior (b).

A neural network model is responsible for sensory-motor coordination and behavior selection. A Hebbian learning mechanism (Hebb, 1949) allows the neural network to adapt during the lifetime of a creature. Additionally an 'artificial biochemistry' is implemented, which models a simple energy metabolism along with a 'hormonal' system that interacts with the neural network to model diffuse modulation of neuronal activity and staged onto genetic development. Both the network architecture and details of the biochemistry for a creature are specified by a variable-length 'genetic' encoding, allowing for evolutionary adaptation through sexual reproduction.

In the game the player is presented with a large fictional landscape. This world contains a range of locations and strange creatures, loosely related (and with considerable poetic licence) to Northern European mythology. The player begins near the long-abandoned home of a family of "Norns" (Small Furry Creatures), where a few eggs, languishing in a broken-down incubator, are almost all that remains of the "Nornir" race. The player can hatch these eggs and interact with the newborn "Norns". The home also contains a few other rooms, including a schoolroom, where the player can attempt to teach the "Norns" a language and other basic ideas and also attempt to learn something of the ancient "Nornir" culture. As food resources are limited, the "Norns" will have to move around, searching for food, warding off enemies and dangers, experiencing various mythical stories on the way, and eventually finding a place to reproduce and further evolve. Users of the system can then exchange their evolved creatures, send them to shared places, and share their stories with other players. Figure 45 shows a collection of creatures, or "Norns", produced by actual game players.



(a) (b)

(c)

Fig. 45 Various images of "Norn" creatures evolved in the "Creatures" software.

5.4.3 - Evolva

Experienced in turning artificial life concepts into commercial products, Latham and Todd (see also Section 5.1.1), who had previously designed the "Organic Art" software package, in 2001 went on to create the "Evolva" software package, an action/role playing combat game where players can control a team of four genetically engineered warriors called "Genohunters". These have to fight against deadly alien parasites that infect a local planet and endanger the virtual galaxy. Players can customize their "Genohunters" by manipulating their genetic code or by absorbing DNA from their victims. Adaptation and mutation allow the "Genohunters" to gain access to more weapons and abilities, and of course the "Genohunters" need to eliminate all enemies to save the planet and the galaxy. The deadly alien parasites themselves also exhibit sophisticated behavior, including a messaging system for communicating with each other as well as mutation capabilities. The "Genohunters", on the other hand, possess the ability of "behavioral cloning" to actually learn and mimic the playing techniques that the player develops. Figure 46 shows images of the entire "Evolva" scene with genetically modified "Genohunters" warding off the evil parasites.

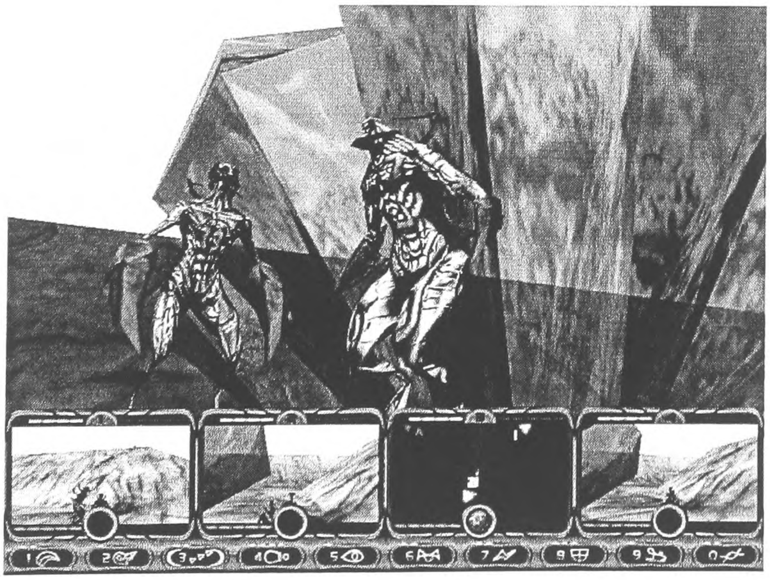


Fig. 46 Screenshot from the "Evolva" game with genetically mutated "Genohunter" creatures.

6 - On-line Artworks Creating Other Forms of Complexity

6.1 - Introduction

In Chapter 5.1–5.4 we have seen artworks, musical systems, design and architectural works as well as game products that apply Artificial Life and Complex Adaptive Systems principles, such as genetic algorithms, genetic programming, and artificial evolution, to production and creation processes. There are generally two distinct approaches to dealing with artificial life: complex adaptive systems and generative processes.

The first group of works uses Genetic Algorithms to create evolutionary still images or 3D shapes, such as the ones described by Todd and Latham (1992), McCormack (1994), Sims (1991, 1993a), Musgrave (1998), Kleiweg (1998), Mount (1998), Gaffney (1998), Damer *et al.* (1999), Rook (1998, 2000), Ishihama (2001), Fagerlund (2001). The Generative Design and Architecture approach by Soddu (2000), Ceccato (2000) and (Pontecorvo, 2000) are also within this group.

The second group models artificial characters or creatures that display autonomous behaviours and use GAs or GP for enabling the artificial evolution of these creatures within their environment. This category includes my own systems (Sommerer & Mignonneau, 1994, 1997a, 1997b, 1998a, 1998b, 1999) as well as the systems described by Ventrella (1996, 1995), Grand *et al.* (1997), Spofford (1998), Heudin (1998), Virtual Fishtank (1998) and Hurry *et al.* (2000).

A third group of works, which I will briefly introduce here, are works that do not explicitly model complex adaptive systems or generative systems but feature properties that can be linked to complex systems, as defined in Chapter 2.2. While these works are not designed as complex system models as such, it is useful to look at them from a "complexity point of view" and determine whether features within these systems emerge that could be considered complex. It is possible that they display several of the following characteristics associated with complex systems: the ability to surprise, dependency, variety, irreducibility, symmetry breaking, emergence, and phase transitions.

During the past several years, artists have become increasingly interested in using the Internet as a platform for modeling interactions between on-line users or between users and artificial on-line characters. The Internet itself consists of an ever-expanding database of images, text and sound files, and it has recently been argued in on-line discussions (see the Complexity On-Line Discussion Forum: http://necsi.org) that the Internet itself could be compared to or even called a Complex Adaptive System. Let us now look at some artists' models which display features of complex systems and try to determine what these emerging features are or appear to be.

6.2 – CollageMachine: Emergence of User Interest Structures through Browsing and Collaging Data from the Internet

One of the first Internet works that dealt explicitly with browsing, through restructuring and recombining complex amounts of data from the Internet, is called "CollageMachine". This system, created by Andruid Kerne in 1997, is a creative web visualization tool that supports open-ended web browsing by deconstructing existing websites into media elements, such as images and chunks of text. While these media elements continuously stream into a collage on the computer screen, a click, drag and drop interface enables the user to rearrange the incoming elements. From this interaction, a software agent learns about the user's interests and acts to shape the ongoing development of the collage on his/her behalf (Kerne, 2001). Figure 47 shows a screenshot of a possible example of this interaction.

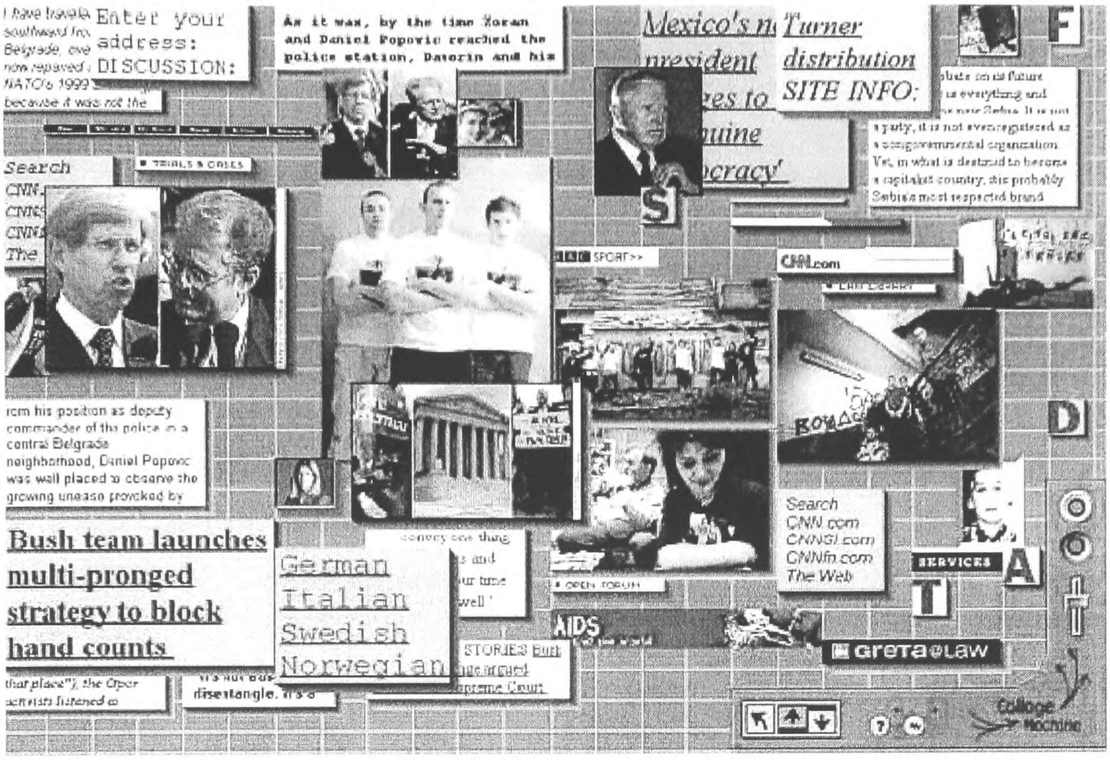


Fig. 47 Screenshot from "Collage Machine" website.

When starting the "CollageMachine," users "seed" the collage session by pointing the system to initial websites that become the sources for the downloads that feed the collage process. From these websites, images and text are cut and pasted onto the "CollageMachine's" Visualization Grid, as shown in Figure XX. Through a control panel interface users can cut, paste, arrange and change the speed of the collage process. The system also includes an agent program that monitors the user's interactions with the various web sites and uses this information to form a model of user interests, which Kerne calls "interest weights".

By assigning a degree of interest, or interest weight, to each of the image or text elements, the system then finds and proposes related image and text information from other websites. Users can choose from these elements and arrange them on screen to create a collage. The size of the image and text icons are directly linked to their internal interest weight, and the amount of "screen real estate" an icon occupies corresponds to the user's interest in a particular topic (Kerne, 2001).

When we evaluate the "CollageMachine" system in terms of the emergence of new structures or the generation of complexity, one of the declared goals of the system is to create "preinventive structures and properties which enable emergence" (Kerne, 2001).

Kerne suggests that the use of inference and indeterminacy within the system can interject ambiguity and incongruity, and he believes that this can "create properties which enable emergence within the system". While Kerne does not exactly demonstrate what these emergent properties might be or how they could be measured, he suggests that the emergent structures are mostly related to the user's perception, that is, how the user might draw connections between certain on- and off-screen information and how the centers of weighted importance might shift and change according to the users preferences. "From what is not there" useful structures may be perceived through the user's subjective gaze, and within the system these appear to emerge; furthermore, a visualization based on the evolving significance weights of certain elements proceeds in a bottom up fashion (Kerne, 2001).

The notion that the appearance of emergent structures ultimately depends on the onlooker's own perception seems to closely adhere to Edmonds definition of complexity as being "relative to the frame of reference" (Edmonds, 1999), as described in Section 2.4.13. As the user's ongoing feedback on the significance of certain information adjusts the weights of the selected images and texts, her/his interests drive the emergent process as much as the user's senses of connection and importance. And since the weights of the images and text information within the "CollageMachine" system are not rendered in a precise fashion, ambiguity and incongruity are embedded in the system.

It is exactly this ambiguity that often intrigues artists. Although this conjecture remains to be tested, such ambiguity and incongruity could be important features of complex systems as they can trigger the ability to surprise (as described in Section 2.4.7).

6.3 – Netomat ™ - Internet Browser

In 1999, a group of artist and programmers created another system which "seeks to transcend the limitations that standard Web browsers and search engines placed on people's interactions with online information" (Wisniewski *et al.*). This system, called "Netomat™", is modeled on the "CollageMachine" system and also aims to reorganize content from the Internet in a non-standard fashion.

Similar to the "CollageMachine" system, its aim is also to treat the Internet as one large distributed application instead of simply a series of linked HTML pages. In "Netomat™" users can browse and interact with text, image and audio data from the Internet in a non-linear fashion and by downloading these image, text or sound files create their own individual collages that represent the amount of interaction and the users' personal preferences. While "Netomat™" represents Internet content in an artistic and "anachronistic fashion" (Wisniewski *et al.*), the system does not seem intended to directly create new or emergent information, except perhaps on a purely visual level.

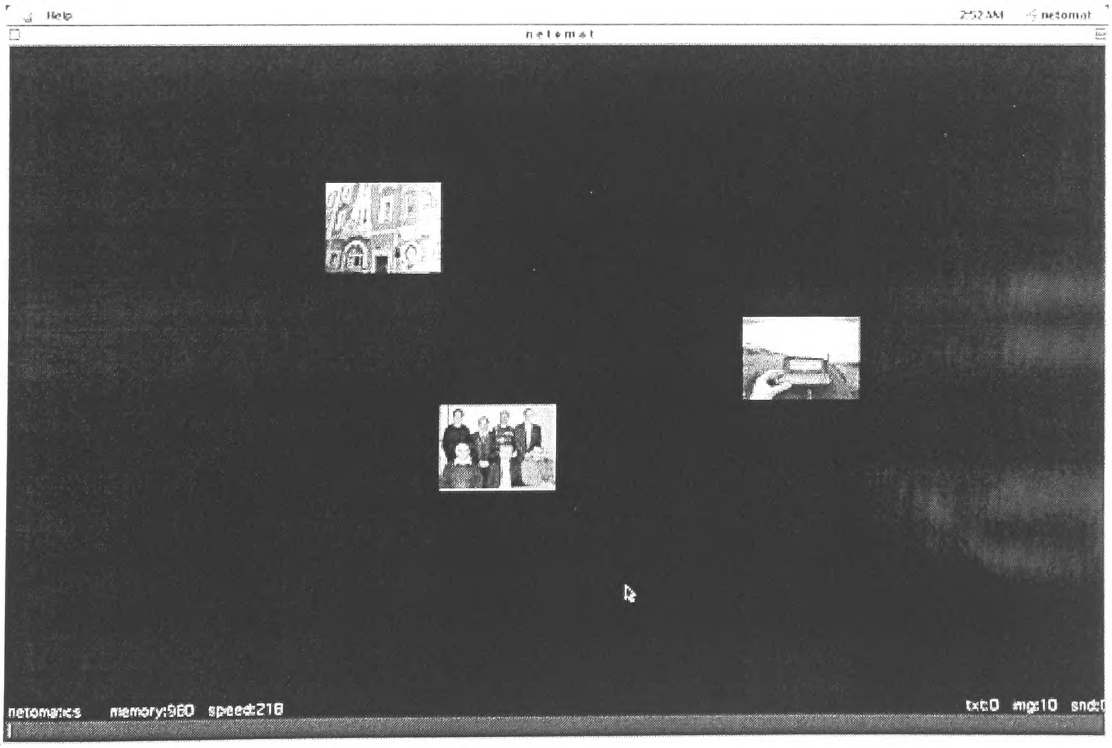


Fig. 47 Screenshot from "Netomat™" website.

Figure 47 shows a screenshot of an image created with the "Netomat™" system; this image appears far less complex and visually rich than the images produced with the "CollageMachine" system.

6.4 - Assoziations-Blaster ! - Interactive Text Network

A step closer to the objective of establishing connections between entities in the dynamically changing Internet is a project called "Assoziations-Blaster". Created in 1999 by two German artists, Alvar C.H. Freude and Dragan Espenschied from the Merz Academy in Stuttgart, it is an interactive text network with automatic and non-linear real-time linking of users' text inputs (Freude *et al.*, 1999). Users here can write single words or whole sentences into the website's graphical user interface and the words within these texts become keywords that link to related texts within a large and ever-expanding database that contains the same keywords. All links between keywords are generated in real time, producing a constantly changing database of texts that link to each other in a non-liner fashion.

Interacting with the "Assoziations-Blaster" could be compared to a collective association session where users generate and find new links and connections between words and their meaning. While users cannot perceive the entire database of texts at once, they can still jump from one word to another to explore the connections between these texts. Figure 48 shows the example sentence "culture is television", and Figure 49 shows one of the associative links the system has proposed when the keyword "television" is chosen. Each underlined word in the text represents a new hyperlink, and users can choose to navigate further into those words and their associated texts.

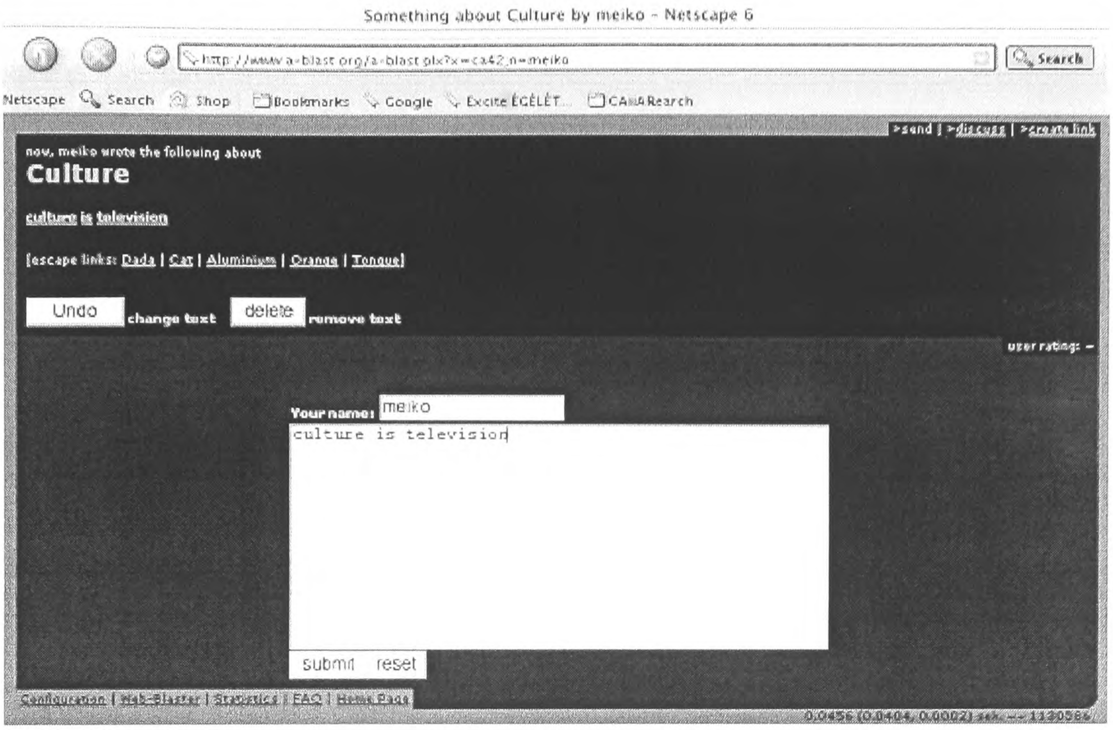


Fig. 48 Screenshot from "Assoziations-Blaster" website and a text input.

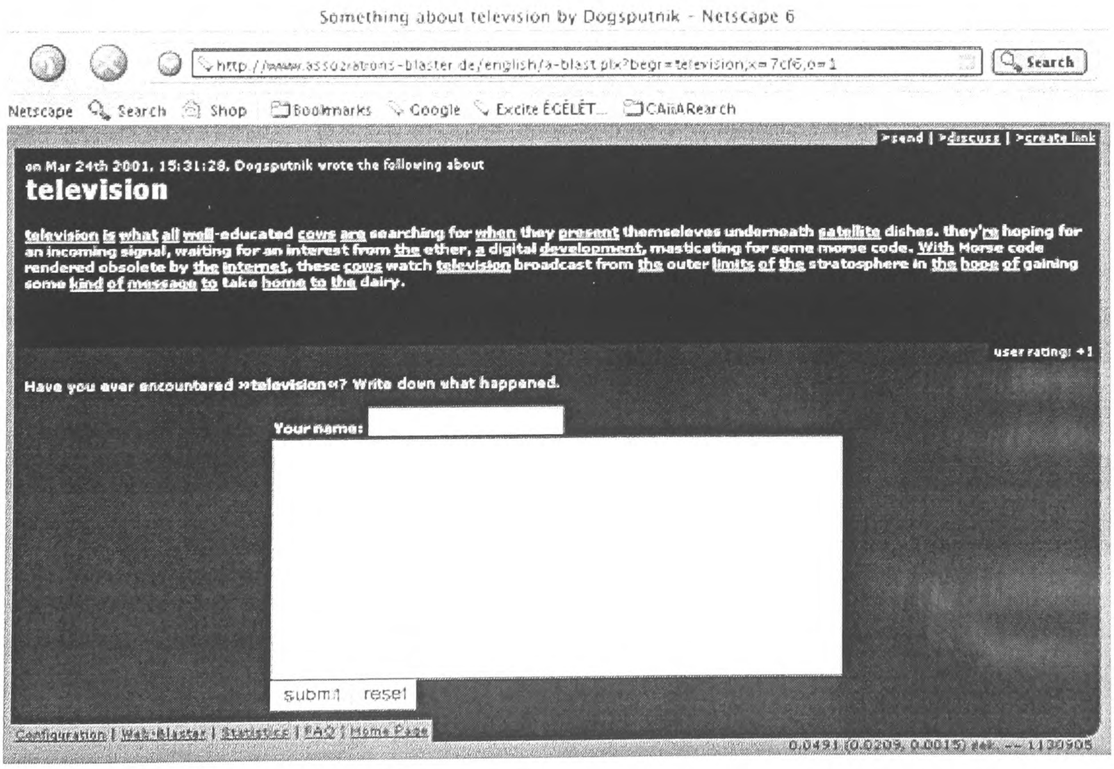


Fig. 49 New text hyperlink generated from original text input shown in Fig. 48.

According to the "Assoziations-Blaster" website information (Freude *et al.*, 1999), every text entry is on the same hierarchical level and not dependent on input time, thread-chains or any other criterion. Automatic links can change the meaning of a single text, and the links themselves change over time as new data is input into the "Assoziations-Blaster!" database. All of the described effects are made transparent on a detailed statistics page, where users can watch how their writings affect the traffic within the search engine, the rating of popular keywords and links, and the overall direction of the entire text structure. An example of such a constantly updated web statistics page is shown in Figure 50.

In addition to inputting new texts and linking them to the existing database, users can also rate the various texts and set a personal threshold if they chose to only see "filtered" texts. While users appear to be in control of the site's content and development, the choice of which word in a sentence structure is actually chosen as a valid keyword to be hyperlinked depends on how many points a users has received through the rating system. This produces only limited control for users who want to engage in interaction only briefly or do not want to collect bonus points.

Coming back to our initial aim of looking at different artworks that qualify as complex and/or emergent systems, "Assoziations-Blaster" appears to be a good example of a non-linear, non-predictable, and constantly changing text structure where dynamic hyperlinks are used to connect pieces of information that had no connection before. Since the user's input directly effects the system's overall development and connectivity, as well as the development of "weights" of interests, one could argue that the entire text structure qualifies as a complex emergent system: it displays several key features such as the "ability to surprise", "variety", "dependency" and, arguably, also "symmetry breaking" (as defined by Heylighen (1996), shown in Section 2.4.9).

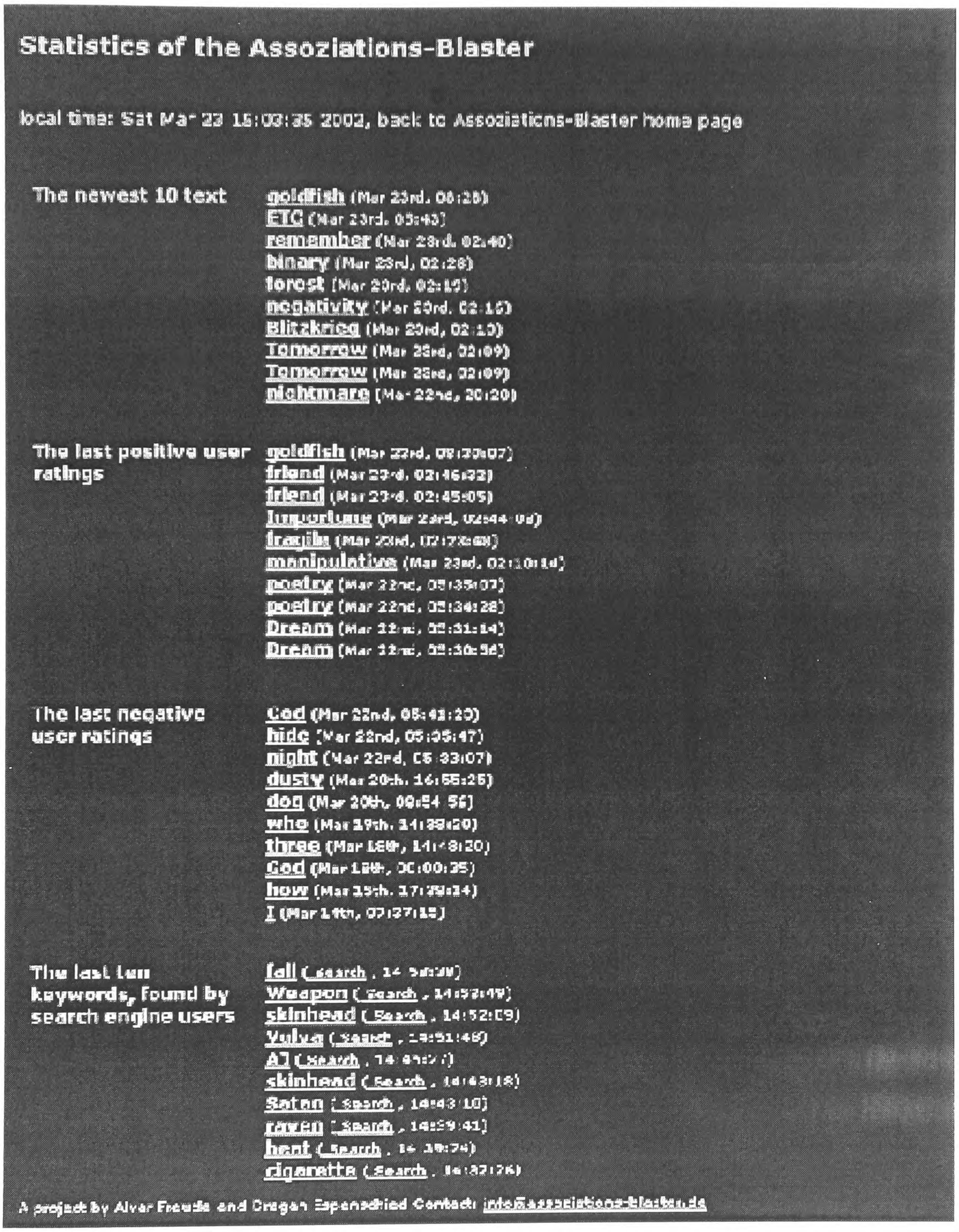


Fig. 50 Example of constantly updated statistics in “Assoziations-Blaster.”

With its internal process of dynamic hyper-linking and weighting of user-driven interaction data, the “Assoziations-Blaster” appears to be similar to the “CollageMachine” system, the only difference being the output modalities, text in the former case and images in the latter. While it can be assumed that the “Assoziations-

Blaster's" internal connectivity will increase over time as more users interact with the system, no studies have been done on what these emergent structures could be and how they could be quantified. The motivation behind the "Assoziations-Blaster" is mostly to create an ever-growing and dynamically linked text database of experimental poetry where fragmented and somewhat random associations between words and texts create partially linked and partially segmented centers of semantic meaning and metaphors. However, the system can also serve as a good example of how emergent systems can be modeled by hyperlinking information and then studying how centers of weight can emerge through this process.

6.5 – Habbo Hotel – Multi-User Chat Environment

A completely different approach to modeling complex interaction between on-line users is the multi-user social environment "Habbo Hotel". Created in 2000 by Finnish artists and designers Sampo Karjalainen and Aapo Kyrölä, the system is an on-line chat room with a sophisticated graphical interface (Karjalainen *et al.*, 2000).

Users or "guests" of the "Habbo Hotel" can move around in different rooms, such as the main lobby, the lido, different bars and cafes, some dancing places, the swimming pool or the kitchen. Rooms are modeled similar to real spaces, where the character needs to avoid objects and can only pass through paths, doors and corridors when they are not occupied by others. Characters in "Habbo Hotel" walk around and choose certain paths by clicking onto target places. Users can stand, walk, sit or dance as well as to talk to other Habbos by turning toward them, either shouting, whispering or speaking normally. A screenshot of the "Habbo Hotel" main page with entrance portal and statistics of user frequency is shown in Figure 51.

Fig. 51 "Habbo Hotel" entrance page with user statistics and entrance portal.

Upon entering the hotel a user has to create his own on-line character (or Habbo) by choosing from different options for its hairstyle, cloths, skin color, etc. The interface used to create new Habbo characters is shown in Figure 52.

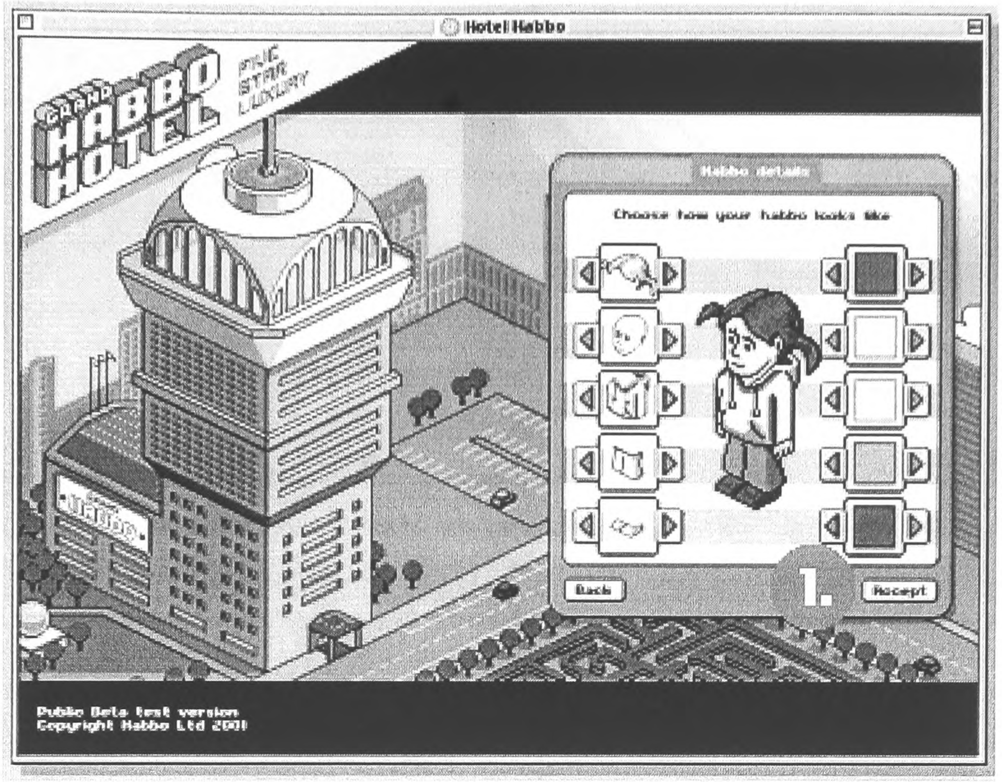


Fig. 52 Interface for designing a "Habbo" character.

When a user wants to engage in a conversation with another Habbo, she simply types a message into a small text editor window. The text then appears as a speech bubble on screen, visible for all the other users in that room. To identify the other characters, the user simply clicks on them and a pop-up menu displays the name and special message of that Habbo character.

Apart from the official "Habbo Hotel" rooms, users can also create their own private rooms, decorate them with furniture they can buy on-line, and invite friends to these rooms. Some rooms can be private but others are open to everyone. The cleverly designed interface of "Habbo Hotel" provides users with direct feedback of their actions, and the conversation and interactions with other Habbo characters create a strong sense of presence and personal exchange. A few examples of private "Habbo Hotel" rooms inhabited by different Habbo characters are shown in Figures 53 and 54.

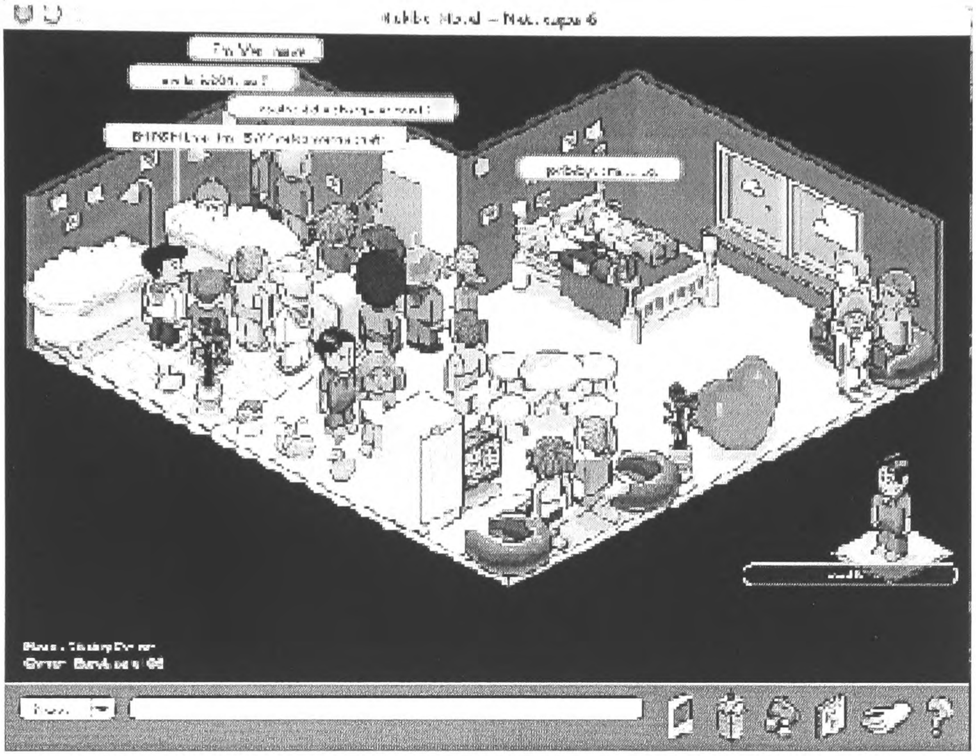


Fig. 53 “Kissing Corner” – private room with Habbos.

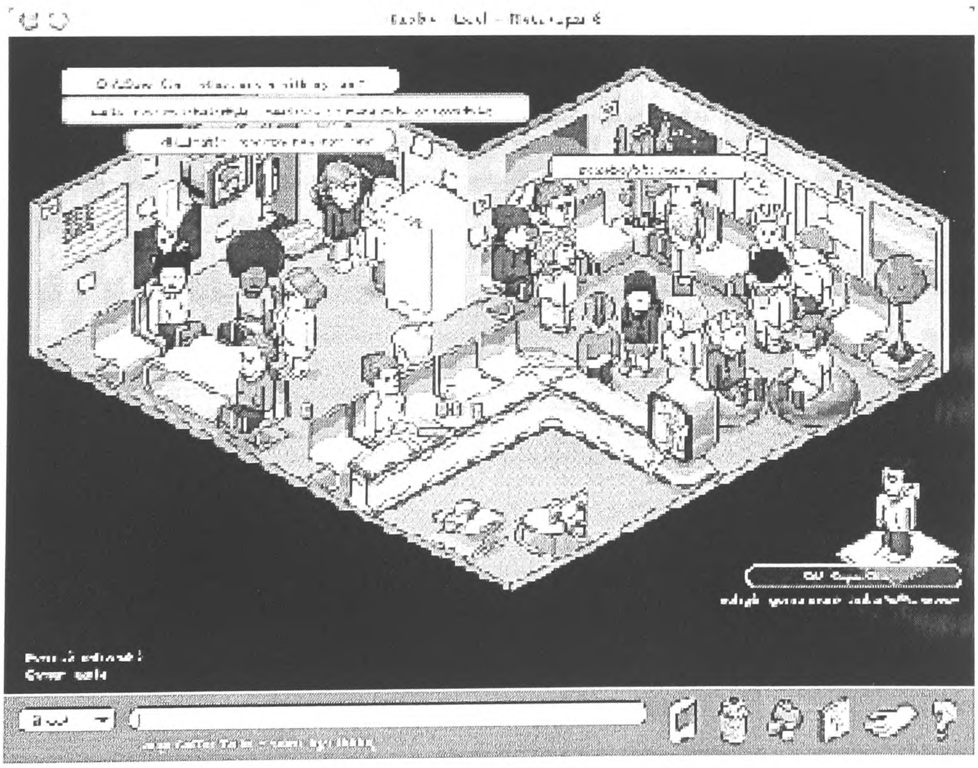


Fig. 54 “Night Club” – private room with Habbos.

In terms of complexity and emergence, the system appears quite interesting because several unprogrammed features such as group behaviours and group activities, the development of a special coded on-line language, and even special character styles and fashions emerge. Since the interaction in "Habbo Hotel" is based on person-to-person encounters and on-line characters are "driven" by real users with complex social experiences, it comes as no surprise that many familiar patterns from our daily social interaction emerge here as well. In one example, some Habbos pretend to be hotel staff, blocking off entrances to certain rooms and not letting other Habbos in. In another case, some Habbos create a private room called "Beauty Comp", where they organized themselves to stage an on-line beauty contest. Here, visiting Habbos are judged on their dancing style, looks, smiles and even conversation style while the "leader" of the room commands the other Habbos to "sit", "dance" or "smile." An example screenshot of this beauty competition with some characters talking to each other and judging each other is shown in Figure 55.

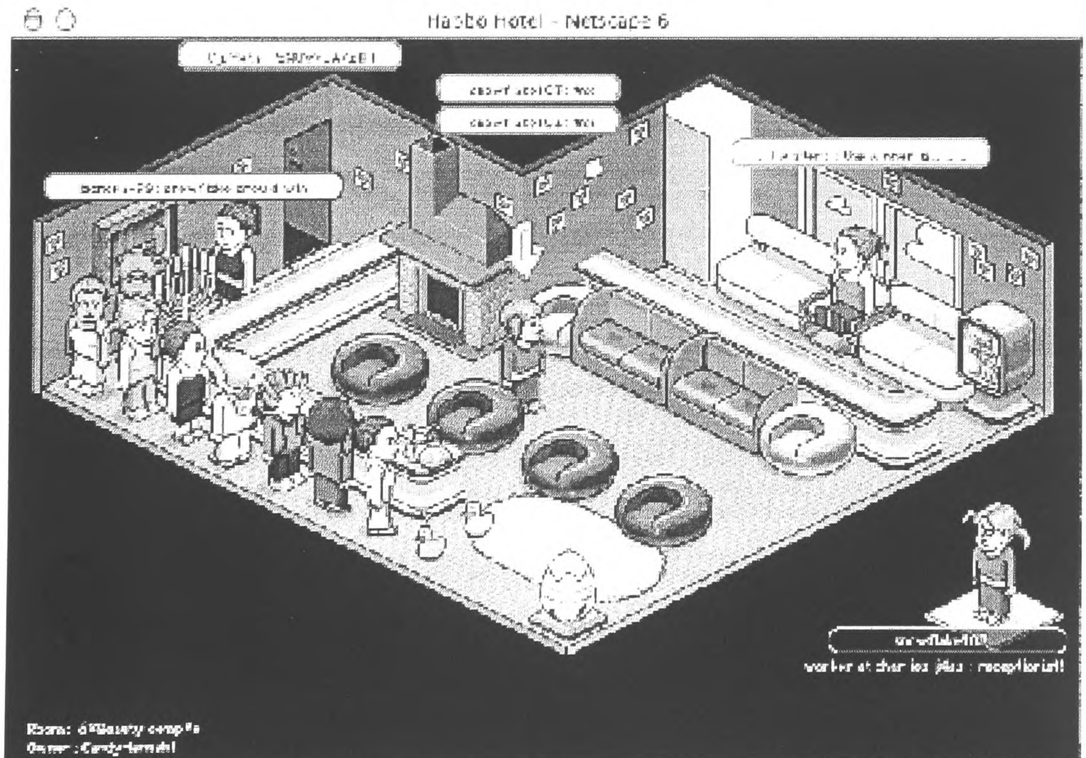


Fig. 55 "Beauty Comp" – privately organized beauty contest held among Habbos.

In another case of emergent behaviour, self-declared Habbo leaders organize Habbo gangs, who start to tyrannize and threaten others. The possibility of trading furniture and goods in exchange for virtual money also helps to create further complex social interactions between the users. Each user can buy objects and then trade them with other Habbos by using the trading interface, shown in Figure 56. According to the "Habbo Hotel" website, (Karjalainen *et al.*, 2000) some cheater characters emerged who pretended to be interested in buying but instead stole objects during the trading.

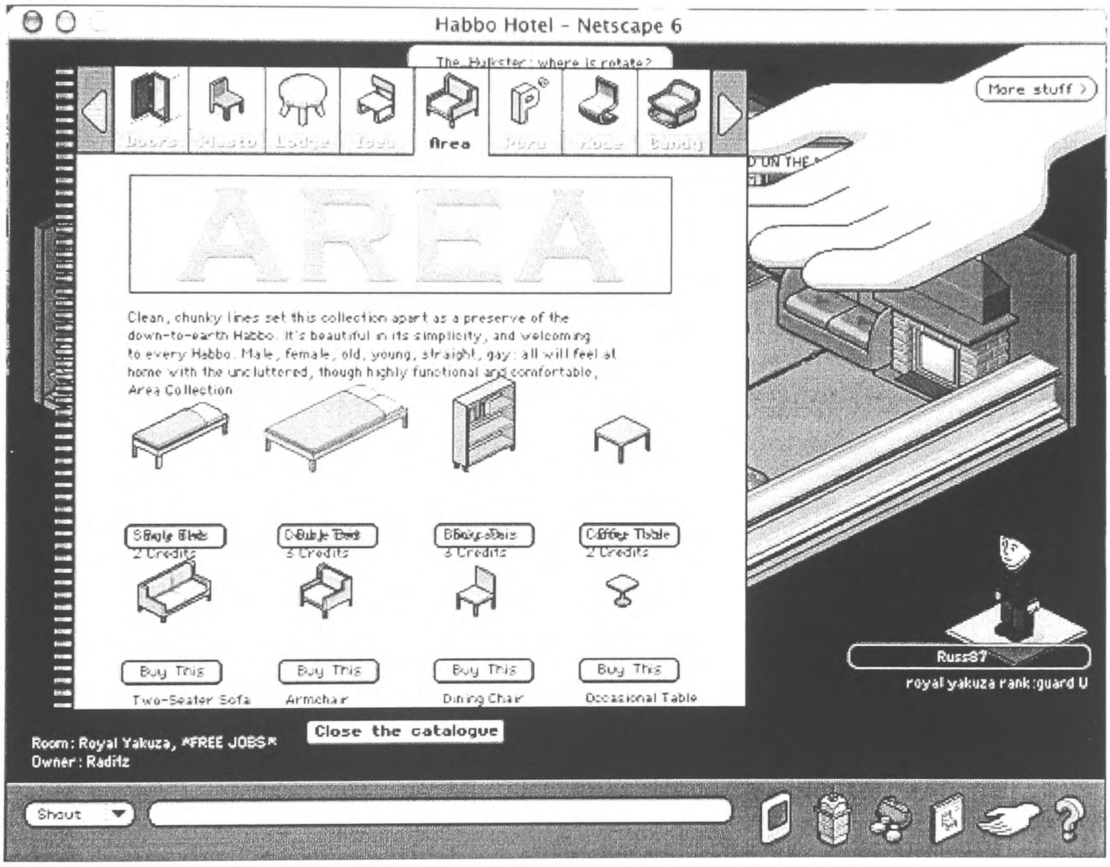


Fig. 56 Buying and trading virtual furniture and items in "Habbo Hotel."

Of course modeling a complex system on-line that taps into already existing social interaction patterns that humans have developed over thousands of years is somewhat easier than trying to model a complex system where all such interactions should emerge bottom-up. But while "Habbo Hotel" transcribes real-life social interactions into the virtual realm, the lesson to learn from this system is that flexible tools (for example, those for the design of one's on-line character or those for various expressions such as chat, whisper, shout, walk, sit, dance, trade) and flexible space organization (such as creating private rooms or specialized rooms for special purposes) are needed to create an open-ended system where new types of interactions, connections and communications between users can emerge.

Another important feature that helps social organization and the building of complex communication structures within "Habbo Hotel" is the fact that users here can always have a global view of where other Habbos are and how they can get in touch with them. The interaction modes in the "Assoziations-Blaster" as well as those in the "Collage Machine" and "Netomat™" systems were mostly designed for user-to-screen interaction with limited possibilities to get in touch with other users and also with a limited overview of the systems' global development. On the other hand, "Habbo Hotel's" focus on user-to-user interaction as well as access to a global view allows users to self-organize more easily and efficiently.

While it could be argued that the social interactions in "Habbo Hotel" are nothing more than a replay of real-life social interactions, the occurrence of certain self-organizing features, such as group building, friendships, gang formations, organizing of events, collective activities or other creative collective behaviours, shows that even limited systems can enable emerging features if they are designed in an open and flexible manner. Moreover, emergence is further encouraged when a system allows single entities (the real and virtual players) to interact with each other in an open, flexible and creative manner while at the same time providing access to important global parameters that can change the system's development.

7 – Modeling Complex Adaptive Systems and Complexity for Interactive Art

7.1 - Objectives and Principles

To summarise, we can see that while there are various definitions of complexity (a comprehensive summary is given in Section 2.3) and a whole list of properties are said to be inherent in all Complex Systems (as described in Section 2.4), there is in fact no unified Complex Systems Theory or a "manual" for how to create a Complex System as such.

Perhaps the most applicable form of complexity is evolutionary complexity, modelled in the form of Complex Adaptive Systems and described by Langton (1989) and Kauffman (1993). Complex Adaptive System simulations have been created by Dawkins (1986), Reynolds (1987), Ray (1991), Yaeger (1994), Holland (1994), and Langton *et al.* (1995). These computational models are typically at the point of maximum computational ability, maximum fitness, and maximum evolvability, at a stage that is often referred to as "life at the edge of chaos" or created through a "phase transition". These models are generally characterized by the following features: they can self-organise, metabolize, self-reproduce, couple to each other, react to their neighbours and to external control, learn and adapt, expand their diversity, explore their options and organise a hierarchy of higher-order structures. These features generally lead to more complex behaviours and structures than the mere parts of the system taken by themselves. While these models are typically inhabited by artificially evolving creatures that follow internal interaction parameters, they usually remain closed toward outside interaction parameters.

7.2 - Goal of the Thesis

The goal of my thesis is therefore to apply principles of Complex Systems and Complex Adaptive Systems to interactive art and to construct artworks that can increase their internal complexity by linking user interaction data to the system's internal software structures. I aim to create dynamic interactive systems that can couple to each other, adapt and organise, mutate and evolve, expand their diversity, react to their neighbours and to external control, explore their options, replicate and organise a hierarchy of higher-order structures. Ideally these artworks should become comparable to Complex Systems by satisfying several of their key properties, as outlined in Section 2.2. The potential benefit of this undertaking is the creation of artworks that are not pre-defined or pre-programmed but instead produce an open-ended variety of image outputs that can increase their complexity as users interact with these systems.

7.3 - Outline of Process

To achieve my goal I have studied the definitions, properties, and principles of Complex Systems (Chapter 2), their connection to the Origin of Life Theories (Chapter 3), and their background in Artificial Life and its connection to Complex Adaptive Systems (Chapter 4). I then went on to look at existing models of Complex Adaptive Systems (Section 4.3) and also introduced several artistic strategies that aim to model such systems for art, design, games, architecture and music (Chapter 5). I also gave a brief overview of artworks that generate or model other forms of complexity by using the Internet as the basis for exploration (Chapter 6).

While reviewing the different artworks and design, architecture and music systems, it became clear that there are three categories of complexity for generative artworks. The first group uses Genetic Algorithms to create evolutionary still images or 3D shapes, such as the ones described by Todd and Latham (1992), McCormack (1994), Sims (1991, 1993a), Musgrave (1998), Kleiweg (1998), Mount (1998), Gaffney (1998), Damer *et al.* (1999), Rook (1998, 2000), Ishihama (2001), Fagerlund (2001), and Greenfield (2000) as well as the Generative Design and Architecture approach by Soddu (2000), Ceccato (2000), and (Pontecorvo, 2000).

The second group models artificial characters or creatures that display autonomous behaviours and use GAs or GP to enable the artificial evolution of these creatures within their environment. This category includes my own systems (Sommerer & Mignonneau, 1994, 1997a, 1997b, 1998a, 1998b, 1999) as well as the systems described by Ventrella (1996, 1995), Grand *et al.* (1997), Spofford (1998), Heudin (1998), Virtual Fishtank (1998), Annunziato (2000), and Hurry *et al.* (2000).

The third group finally appears to model complex system in a broader sense, without explicitly focusing on complex adaptive systems or without actually stating an explicit connection to the field of Complex Systems Sciences. Instead, these works use the idea of linking single entities that can create a network of higher-ordered structures, and they all use the Internet as the basis of exploration. The most successful example in this category is the Assoziations-Blaster - Interactive Text Network system (Freude *et al.*, 1999). The Habbo Hotel system (Karjalainen *et al.*, 2000) could also qualify as a complex system, since the interactions between the user-driven Habbo Hotel characters (Habbos) can enable the emergence of novel social features. Compared with the bottom-up approach of trying to model the emergence of novel features within a system (as discussed in Sections 4.3 and 5.1), this top-down approach appears easier, as it relies on already existing knowledge of social behaviours and the resulting complex social interactions. It remains to be tested whether these works are in fact relevant to a discussion on complex systems and, if so, how such complex and novel behaviours could emerge in a stand-alone artificial system without human intervention or preconception. The capability to achieve the complexities and emergent properties of human social behaviours will remain a critical challenge in developing artificial systems that incorporate complex interactions in modelling emergent properties among artificial entities.

7.4 - Outline of Proposed Systems

As stated in Section 1.2, the objective of my thesis is to investigate the issue of Complexity through a two-fold path. First I will test the possibility of modelling a Complex Adaptive System for Interactive Art and see how this process can create image and software structures that can become increasingly complex as users interact with these systems. To accomplish this, I will program software structure that is inhabited by three-dimensional artificial creatures. These creatures are able to move, use a primitive vision system, metabolise, compete for energy, look for mating partners to reproduce, and pass on their genetic material to the next generation. This can be achieved by applying Genetic Algorithms (GA) and Genetic Programming (GP) (as described in Sections 4.2.5 and 4.2.6) to 3D computer graphics. Similar evolutionary software models, called Complex Adaptive Systems, have already been described by Dawkins (1986), Yaeger (1994), Ray (1991), and Holland (1994). My proposed model adds to this body of work by introducing a new feature: user interaction is a key factor in the creatures' design as well as in their overall behaviours, interactions and survival within the artificial world (Sommerer & Mignonneau, 1997a, 1997b). The following parameters are used to design this system:

- creature's shape is related to its genetic code
- user designs the creature's shapes (genetic code) through human-computer interfaces
- creature's shape decides its movement capability (speed)
- creature's fitness depends on its speed
- creature's energy level changes with movement
- creature's interaction decisions depend on fitness and energy
- creatures can mate and reproduce if sufficient energy and fitness are available
- child creatures inherit the genetic code of their parents (using GP techniques)
- creatures sense the internal environment through artificial vision
- creatures sense the external environment through human-computer interfaces
- creature's interaction decisions depend on internal and external parameters
- creatures and users react to each other through human-computer interfaces

Based on these objectives, I will introduce a system called Life Spacies II. This work was commissioned by the ICC-NTT InterCommunication Museum in Tokyo, and the first version, called "Life Spacies", was shown in spring 1997. In the following chapter I will describe these two systems in depth and analyze their complexity potential.

For the second path of my investigations, I decided to explore other forms and properties of complexity by using the Internet as a large and dynamic database that can be used in interactive artworks. Here, my objective is to create an interactive system where users can interact with complex image and sound data and in the process create new connections, new combinations, and new relations that can construct an emerging and complex system that is no longer reducible to its mere parts. As we have seen in Chapter 6, during the past several years artists, including myself, have become increasingly interested in using the Internet as a platform for modelling interactions between on-line users or between users and on-line characters. The Internet itself consists of an ever-expanding database of images, text and sound files, and it has recently been argued that the Internet itself could be called a Complex System or compared to a Complex Adaptive System.

To expand the notion of Complex Systems and to explore new territory, I therefore designed several on-line systems that use the Internet as a platform for experimentation and for modelling complex interaction patterns. These systems rely on the unpredictability of image, sound and text information, and as users interact with this information, new links, new connections and new contextual information are created. While I am aware that these features are far less quantifiable than the complexity measure outlined in Section 2.2, these systems nevertheless seem to show several of the other features associated with complex systems as defined in Section 2.4: variety, dependency, irreducibility, ability to surprise, connectivity, symmetry-breaking, and some form of emergence, perhaps relative to the frame of reference. In the following two chapters I will introduce my artistic systems in more detail.

8 - Life Spacies II: Complex Adaptive Systems for Interactive Art

8.1 – Introduction

As stated in Chapters 2 and 7, the aim of this thesis is to apply principles of Complexity and Complex Adaptive Systems to Interactive Art. We would now like to introduce our own artistic interpretation of a Complex Adaptive System, called the "Life Spacies" system and it's follow-up version "Life Spacies II."

8.2 - Life Spacies

"Colourless green ideas sleep furiously" (Noam Chomsky)
According to Noam Chomsky, human language acquisition is based on a universal grammar that is genetically embedded within the human mind of all normal children, allowing them to learn their native languages naturally and seemingly effortlessly (Chomsky, 1972). It was also Chomsky who coined the above phrase of "colourless green ideas sleep furiously," an expression that might not make much logical sense to a more scientifically oriented person, but does have quite a lot of meaning for a more visually or artistically minded person. Though this sentence, as Chomsky has shown, is grammatically correct, its meaning cannot be grasped through logic alone. When we hear this sentence for the first time we see pictures or forms or shapes appearing in our minds. These forms are vague, yet they are defined to a certain degree and can certainly create visual sensations and emotions. Inspired by Chomsky's sentence and based on the idea of translating words or sentences into visual forms, we created an interactive system called "Life Spacies". This system was published in the literature (Sommerer *et al.*, 1997c, 1998a, 1998b, 1999a, 1999b, 1999c, 1999d, 1999e, 1999f).

8.2.1 - Background

Based on the principles of evolutionary design linked to multi-model interaction, "Life Spacies" was produced for the ICC InterCommunication Museum in Tokyo as part of its permanent collection (Sommerer & Mignonneau, 1997c). It is an interaction and communication environment where remotely located visitors on the internet (i.e., global environment) and on-site visitors to the installation at the ICC Museum in Tokyo (i.e., local environment) can interact with each other through artificial creatures. Artificial creatures are created by on-line participants through writing email messages to the "Life Spacies" web page. Each text message provides the genetic code for a creature, and a specifically designed text-to-form editor allows us to translate text into form. The following sections describe this text-to-form editor in more detail and explain the system's set-up and the interactions between creatures and visitors in the "Life Spacies" interaction environment.

8.2.2 - System Description

"Life Spacies" is an evolutionary communication and interaction environment that allows remotely located visitors to interact with each other in a shared virtual environment. Visitors can integrate themselves into a three-dimensional complex virtual world of artificial life organisms that react to their body movement, motion and gestures. These artificial beings also communicate with each other as well as with part of an artificial universe, where real and artificial life are closely interrelated through interaction and exchange. Through the "Life Spacies" web page (Figure 57), people all over the world can contribute to the system by simply typing and sending an email message to the "Life Spacies" web site (http://www.ntticc.or.jp/~lifespacies) to create one's own artificial life form.

These creatures will immediately start to live in the "Life Spacies" environment at the ICC museum and interact with the visitors on-site. After sending the email message, people will receive a small curriculum vitae for their creature, as well as an image of how it looks. The artificial species can be created in two different ways:

a) through incoming international email messages. A text-to-form editor creates the genetic code for each creature:
- one message is one creature
- complex text messages create complex creatures

- different levels of complexity within the text correspond to different kinds of species

b) by the creatures themselves:

- reproduction helps the creatures to propagate their genotype in the system so they can form groups of different species

After sending an email message to the "Life Spacies" web page, the sender also soon receives a curriculum vitae for his creature, as well as an image of how it looks. When the creature dies, a report is given to its creator, telling him or her how long the creature lived and how many children and clones it produced.

"Life Spacies" is again based on the idea of evolutionary design, which is not predetermined by the artist but solely dependent on the interaction of the visitors and the evolutionary process. Only the messages mailed from people all over the world and the reproduction and evolution of the creatures themselves will determine how the creatures look and behave. Consequently, one cannot really predict how the work will evolve and what kind of creatures will emerge. It will exclusively depend on how many people send messages, how complex these messages are, and how the creatures reproduce among themselves and through the selection of the visitors in the museum.

8.2.3 - Life Spacies' GUI

The "Life Spacies" graphical user interface (GUI) consists of a web page, as shown in Figure 57. It allows people throughout the world to interact with the system. By simply typing and sending an email message to the "Life Spacies" web site (http://www.ntticc.or.jp/~lifespacies), one can create one's own artificial creature: this creature starts to live in the interaction environment at the ICC Museum as soon as the message is sent. To achieve this process, we developed a special text-to-form editor that enables us to translate text into genetic code.

Fig. 57 "Life Spacies" web site.

8.2.4 - Text-to-Form Editor

The text-to-form editor is based on the idea of linking the characters and syntax of a text to specific parameters in the creature's design. The default form of a creature is a body made up by a sphere consisting of 100 vertices, 10 rings with 10 vertices each. All vertices can be modified in x, y and z axes to stretch the sphere and create new body forms. Several bodies can be attached to each other provided that their attachment point is located on the x-axis. If the attachment point is not on the x axis, a limb is created instead of a body; this limb is copied and the copy is attached at a position symmetric to the original position. Figure 58 shows a creature with two spheres as bodies and one pair of limbs.

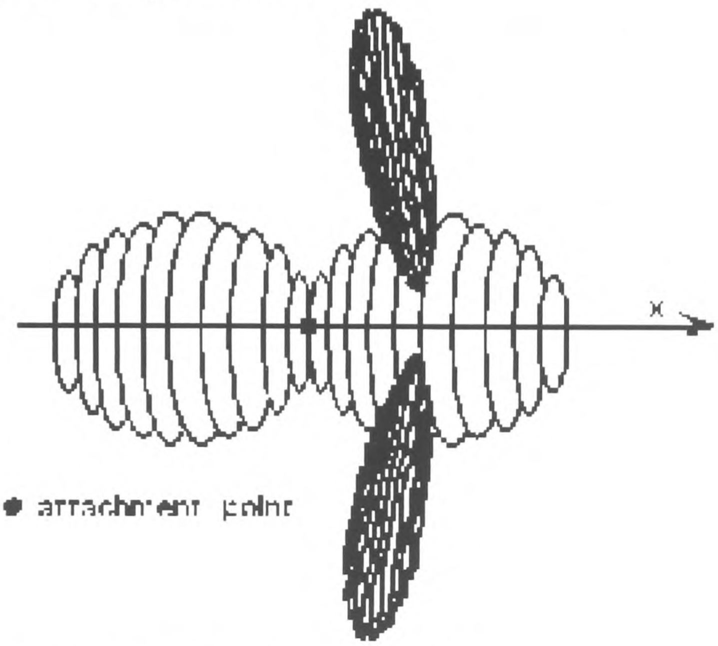


Fig. 58 Creature with two bodies and one pair of limbs.

According to the sequencing of the characters in the text, the parameters of x, y and z for each of the 100 vertices can be stretched and scaled, the colour values and texture values for each body and limb can be modified, the number of bodies and limbs can be changed, and new locations for attachment points of bodies and limbs can be created. Since each of the vertex parameters is changeable and all of the bodies and limbs can be changed as well, about 50 different functions for the creature's design parameters are available. They are listed in a design function table (Figure 59).

function1 stretch default body/limbs in x
function2 stretch default body/limbs in y
function3 stretch default body/limbs in z
function4 set the next stretch function to global
function5 set the next stretch function to a vertex point
function6 set the next stretch function to a ring
function7 create a new location for an attachment point
function8 copy a new location for an attachment point
function9 compose a new texture for body/limbs

function10 copy texture of body/limbs

function11 change parameters of RED in body/limbs texture

function12 change parameters of GREEN in body/limbs texture

function13 change parameters of BLUE in body/limbs texture

function14 change patterns of body/limbs texture

function15 exchange positions of bodies/limbs

function16 copy body/limbs

function17 create a new body/limbs

function18 add or replace some of the above functions

function19 randomize the next parameters

function19 copy parts of the previous operation

function20 modify life span (default is 24 hours)

function21 add the new parameter to previous parameter

function22 ignore the current parameter

function23 ignore the next parameter

function24 replace the previous parameter by new parameter

....

function50

Fig. 59 Design function table.

Next, in translating the characters of the text message into these design function values, we first assign an ASCII value to each character. This is done according to the standard ASCII table as shown in Figure 60.

33 !	34 “	35 #	36 $	37 %	38 &	39 ‘	
40 (	41)	42 *	43 +	44 ,	45 -	46 .	47 /
48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7
56 8	57 9	58 :	59 ;	60 <	61 =	62 >	63 ?
64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G
72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W
88 X	89 Y	90 Z	91 [	92 \	93]	94 ^	95 _
96 `	97 a	98 b	99 c	100 d	101 e	102 f	103 g

104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w
120 x	121 y	122 z	123 {	124 |	125 }	126 ~	

Fig. 60 ASCII table.

We see that each character refers to an integer. We can now proceed by assigning this value to a random seed function *rseed.* In our text example from Figure 61, *T* of *This* has the ASCII value 84, hence the assigned random seed function for *T* becomes *rseed(84).* This random seed function now defines an infinite sequence of linearly distributed random numbers with a floating-point precision of 4 bytes (float values are between 0.0 and 1.0). These random numbers for the first character of the word *This* will become the actual values for the modification parameters in the design function table. Note that the random number we use is a so-called "pseudo random," generated by an algorithm with 48-bit precision, meaning that if the same *rseed* is called once more, the same sequence of linearly distributed random numbers will be called. Which of the design functions in the design function table are actually updated is determined by the following characters of the text, i.e., *his*; we then assign their ASCII values (104 for *h*, 105 for *i*, 115 for *s* ...), which again provide us with random seed functions *rseed(104)*, *rseed(105)*, *rseed(115).* These random seed functions are then used to update and modify the corresponding design functions in the design function look-up table, between design function1 and function50. For example, by multiplying the first random number of *rseed(104)* by 10, we get an integer, which assigns the amount of functions that will be updated. Which of the 50 functions are updated is decided by the following random numbers of *rseed(104)* (as there are 50 different functions available, the following floats are multiplied by 50 to create integers). Figure 45 shows in detail how the entire assignment of random numbers to design functions operates. As mentioned above, the actual float values for the update parameters come from the random seed function of the first character of the word, *rseed(84).* An example of the entire procedure is given in Figure 61.

Example word: *This*

T => *rseed(84)* => {0.36784, 0.553688, 0.100701,...}

(actual values for the update parameters)

h => *rseed(104)* => {0.52244, 0.67612, 0.90101,...}

0.52244 * 10 => get integer 5 => 5 different functions are called within design function table

0.67612 * 50 => get integer 33 => function 33 within design function table will be updated by value 0.36784 from 1. value of *rseed(84)*

0.90101 * 50 => get integer 45 => function 45 within design function table will be updated by value 0.553688 from 2. value *rseed(84)*

.... until 5. value

Fig. 61 Example of assignment between random functions and design functions.

As explained earlier, the basic "module" is a sphere, with a default colour white and no texture. When messages are sent, the incoming text modifies and "sculpts" this default module by changing its form, size, colour, texture, number of bodies/limbs, copying parts and so forth. Depending on the complexity of the text, the body and limbs of the creature become increasingly shaped, modulated and varied. As there is usually great variation among the texts sent by different people, the creatures themselves also vary greatly in appearance, thus providing a personal creature for each author. Figure 62 shows an example of a short and simple email message sent to the "Life Spacies" web site.

Date: Sun, 01 Nov 1998 13:14:32 +0900

From: Christa Sommerer <christa@mic.atr.co.jp>

To: life@lc.ntticc.or.jp

CC: christa@mic.atr.co.jp

Subject: test creature1

This is a test creature.

Fig. 62 Example of email message to “Life Spacies”.

8.2.5 - Picture of Life Spacies Creatures

To provide feedback to the on-line users after their messages are sent, users receive an image as well as a curriculum vitae of their creatures. As soon as a message is sent to the server in Tokyo, the creature starts to live in its virtual environment and the author of the text receives a picture of his or her creature in return.

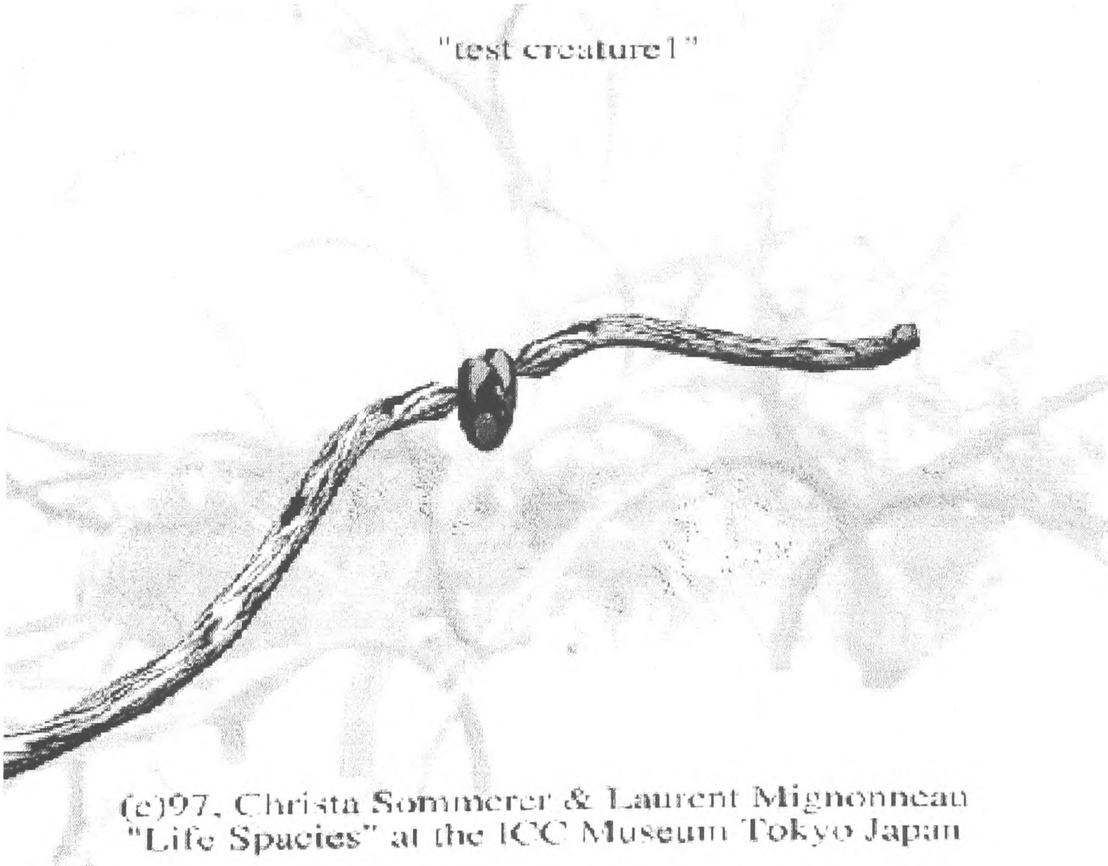


Fig. 63 Creature created by email in Figure 62.

Figure 63 shows an image of the creature created by the text message of Figure 62. Because the text message was rather short, the corresponding creature consists of just one body and one pair of limbs, similar to the default case but with long limbs and a heart-shaped body.

8.2.6 - Variations in Creature's Design

By only changing the first character of each word, a different random seed is chosen for the following characters of the word and, consequently, the design for body and limbs will change. Figures 64 and 65 show the effects of changing the original message of Figure 46 by modifying the *H* into *F*, the *T* into *R*, *i* into *o*, *t* into *n* and *c* into *g*.

```
Date: Sun, 01 Nov 1998 13:15:07 +0900
From: Christa Sommerer <christa@mic.atr.co.jp>
To: life@lc.ntticc.or.jp
CC: christa@mic.atr.co.jp
Subject: test creature1var2

Rhis os i next greature.
```

Fig. 64 Modified email message.

We see that the new creature in Figure 65 still consists of one body and one pair of limbs, but its form, size, orientation and colour of body and limbs have changed significantly.

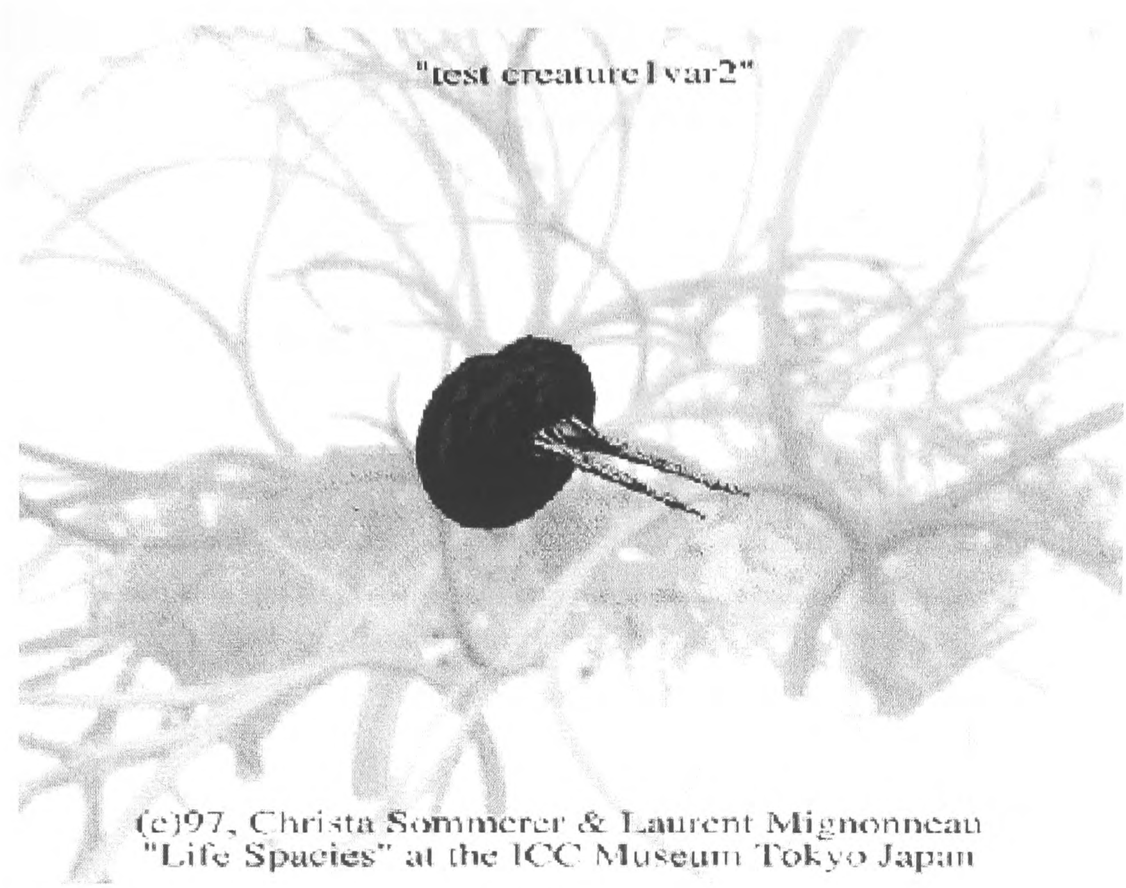


Fig. 65 Creature created by modified email message.

When more complex messages with more characters, words and varied syntax are sent, more elaborate creatures with more bodies, limbs and variation in body form, texture, size and colour can be created. Figure 66 is a much more detailed text message and Figure 67 shows the resulting creature's image.

Date: Sun, 01 Nov 1998 13:20:32 +0900
From: Christa Sommerer <christa@mic.atr.co.jp>
To: life@lc.ntticc.or.jp
CC: christa@mic.atr.co.jp
Subject: example #4

this is not a sentence, it is a creature, it is now in Tokyo, where it lives.
it is a creature, this is not a sentence, where it lives, it is now in Tokyo.
it is now in Tokyo, this is not a sentence, it is a creature, where it lives.

Fig. 66 Complex email message.

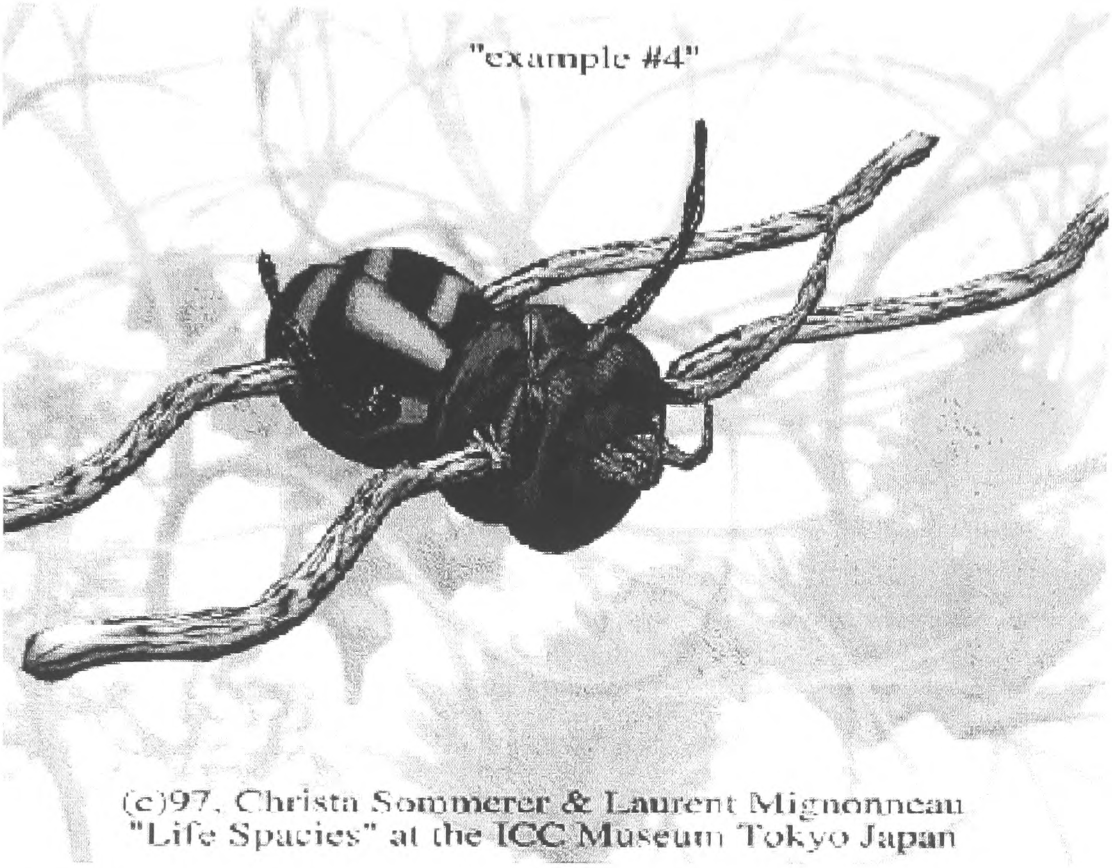


Fig. 67 Creature created by complex email in Figure 66.

8.2.6 - Curriculum Vitae of Creatures

A creature's default life span is 24 hours, but as the life span is also a function determined by the design function table (Figure 45), it is updated and changed through the values of the specific characters in the text. When the creature dies, a report is given to its author, telling him or her how long the creature lived and how many children and clones it produced. Figure 68 shows the Curriculum Vitae of the creature in Figure 67.

From: "Life Spacies" <life@lc.ntticc.or.jp>
Date: Thu, 6 Aug 1998 00:00:15 +0900
Subject: Curriculum Vitae of Creature "example #4"
To: christa@mic.atr.co.jp

Hello Christa,

This is an automatic email message from Life Spacies.

Here is the curriculum vitae of your creature

called "example #4" from the ICC Museum

```
*-------------------------------------------------------*
|                 Curriculum-Vitae                      |
*-------------------------------------------------------*
```

Born in Tokyo, Japan on :

- (japan time) Tue Aug 4 00:41:21 1998

the Creature "example #4"

has got 13 clones and 2 kids.

It has been moved 18 times away from its habitat

and has been touched 14 times by the ICC visitors.

The creature "example #4" was living until:

- (japan time) Wed Aug 5 00:00:08 1998

your email text was setting a lifespan of :

- 0 days 23 hours 18 min.

Fig. 68 Curriculum Vitae of creature in Figure 67.

8.2.8 - On-site Interaction

On-site interaction took place at the ICC InterCommunication Museum in Tokyo, where the "Life Spacies" system has been exhibited as part of the museum's permanent collection since 1997 (Sommerer & Mignonneau, 1997c).

8.2.9 - Interaction Setup

The interaction setup consists of two independent interaction sites (Figure 69) that are linked together via a data line, allowing visitors at remote locations to be displayed and interact in the same virtual three-dimensional space. The system setup is based on earlier interactive installations called "Trans Plant" (Sommerer & Mignonneau, 1995) and "MIC Exploration Space" (Sommerer & Mignonneau, 1996).

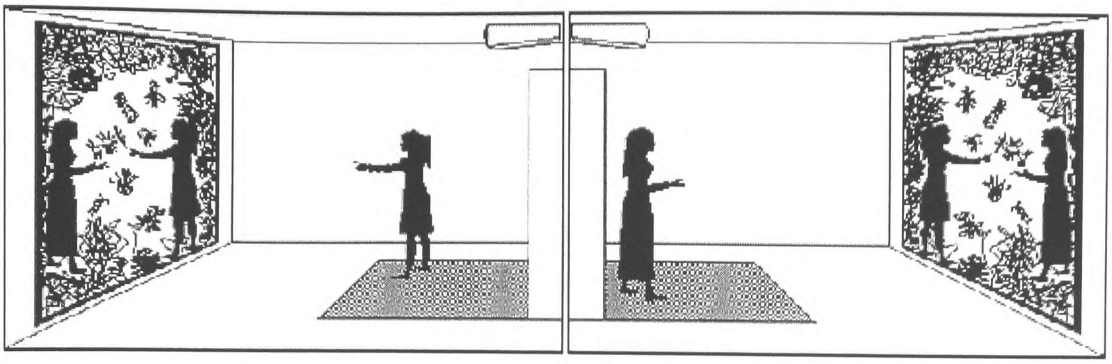


Fig. 69 Life Spacies Interaction Setup.

8.2.10 - Interaction between Users and Creatures

On-site visitors can directly interact with the creatures through touching and catching them. Once a creature is caught by the visitor, it will clone itself. However, if two remotely located people are in the same virtual space, they can each catch a creature with their hands, which causes these two creatures to mate and create an offspring by genetically exchanging the parents' code.

8.2.11 - Cloning Creatures

If the visitor catches a creature, it makes a perfect copy, or clone, of itself. The creatures are basically shy, and one needs to look for them carefully because they hide in the branches of vegetation.

8.2.12 - Mating Creatures

When the two remotely located visitors each catch a creature, these two creatures mate and create an offspring, a child creature. In this case, the offspring inherits the genetic code of the parent creatures; this is done through cross-over of the parents' codes and application of minimal mutation. Cross-over can take place at any part of the genetic string (i.e., text) of the creatures; the location and length of the cross-over is decided at random, but it is adapted to the length of the genetic string (i.e., text) of the creatures. Figure 70 shows an example of a genetic exchange through cross-over and mutation.

```
Parent creatures (1) and (2), child creature (3);
| ....  indicates the area of cross over;
^ ....  indicates the location of mutation;


(1) This is a crea|ture, it| lives in Tokyo.
              ^    |      |          ^
(2) This creature |is now l|iving in Tokyo.
                  |       |

(3) This is ancrea is now l lives in Toky .
```

Fig. 70 Cross-over and mutation to create child creature.

8.2.13 - Interaction and Evolution

"Life Spacies" is a system where interaction and exchange happens between real life and artificial life on human-human, human-creature and creature-creature levels. Aided by our genetic text-to-form editor, users on the Internet can create artificial creatures by writing text messages to the "Life Spacies" web site; additionally, on-site visitors to the installation influence the creatures' reproduction by touching them with their hands and thus promote the propagation of specific gene pools of creatures in the "Life Spacies" environment. "Life Spacies" is again based on the idea of evolutionary design, which is not predetermined by the artist but solely depends on the interaction of the visitors and the evolutionary process (Sommerer & Mignonneau, 1998b). Only the messages mailed from people all over the world and the reproduction and evolution of the creatures themselves determine how the creatures will look and how they will behave. One can thus not really predict how the work will evolve and what kind of creatures will emerge. This will exclusively depend on how many people send messages, how complex these messages are, and how the creatures reproduce among themselves and through the selection of the visitors in the museum. In-depth descriptions of this system were also published in (Sommerer *et al.*, 1997c, 1998a, 1998b, 1999a, 1999b, 1999c, 1999d, 1999e, 1999f).

8.3 - Life Spacies II

8.3.1 - Introduction

In 1999, I created a stand-alone version of "Life Spacies" called "Life Spacies II." In this system users at the exhibition venue can directly create creatures on-site and watch their creatures be instantly translated into visual three-dimensional forms that interact with each other on a large projection screen. This system also has the advantage that users can see all the creatures simultaneously on the screen and watch their interactions and evolution more clearly. The following sections describe the "Life Spacies II" system in more detail. In-depth descriptions of the system were also published in (Sommerer *et al.*, 1999b, 1999c, 1999d, 1999e, 1999f, 2000a, 2000b, 2000c, 2000d, 2001b).

8.3.2 - System Description

"Life Spacies II" consists of a graphical user interface (GUI) that allows users to type text messages into the Internet web page text editor (Figure 71). As in Life Spacies, written text is used as genetic code to create three-dimensional creatures (Sommerer & Mignonneau, 1999c). Once a text has been written, a corresponding creature will appear on a large 4 x 3 meters projection screen where it will start to move around and look for food.

In addition to creating creatures, users can feed their creatures by releasing text characters through the GUI. Food particles are in fact text characters, and the user can decide how much text, which type of text, and where to place it within the projection screen. Once released, the text (food) instantaneously appears on the screen and will be picked up the creatures (Figure 72).

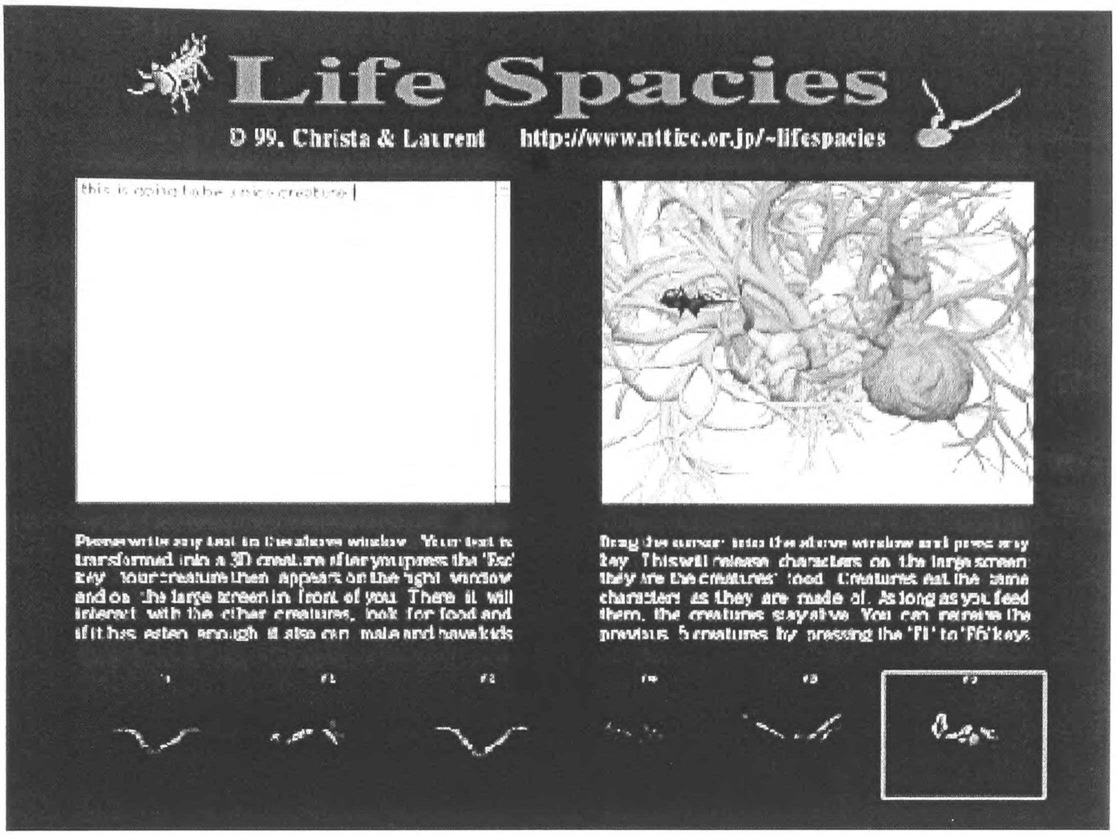


Fig. 71 "Life Spacies II" (GUI). The upper-left window is used to type messages and create creatures, and in the upper-right window text can be released to feed creatures.

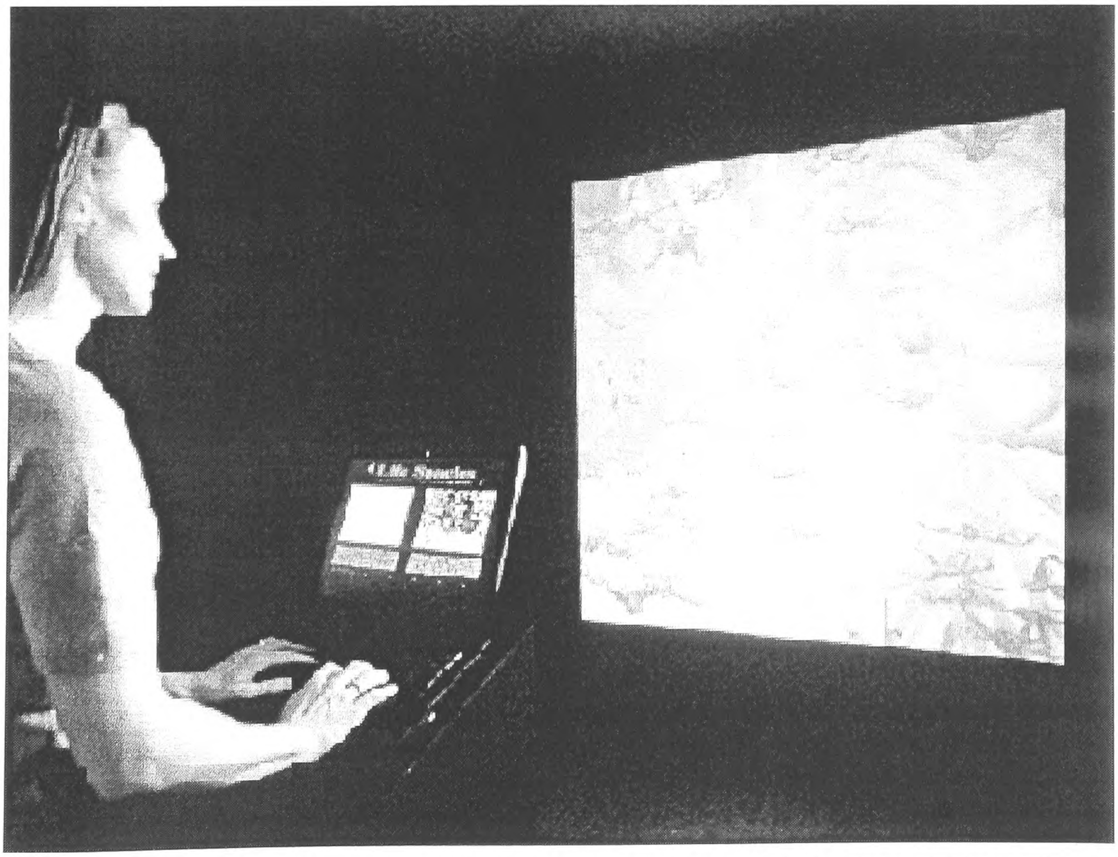


Fig. 72 "Life Spacies II" - user as she creates and feeds creatures through the GUI.

8.3.3 - Behaviour of Creatures

8.3.3.1 - Energy and Speed

A creature's behaviour is basically dependent on two parameters:

a) its Energy level (E)

b) its Speed (S) or ability to move

While the Energy level (E) is a value that constantly changes as the creature moves in its environment and decreases by increased movement, the Speed (S) value is designed by the creature's body physics. A creature with a large body and small limbs will typically move slower than a creature with a small body and long limbs. Additionally, the shapes of the creature's body and limbs have an influence on its ability to move. On the other hand, the Speed (S) value is decided at creation through the text characters' arrangement in the creature's genetic code, which is interpreted and translated by the design function table as described in Section 8.2.4. The creature's behaviour decision parameters are shown in Figure 73.

Speed (S): depends on creature's body and limb size
decides how fast the creature can move
Energy (E): E = 1 at birth
Speed (S) of movement reduces E
E < 1 ..... creature becomes hungry
E > 1 ..... creature can mate

Fig. 73 Creature's behaviour decision parameters.

8.3.3.2 - Interaction Decision Parameters

Based on how much Energy (E) a creature has at a given moment and how fast it can move in the environment (Speed (S)), the creature's interaction with other creatures will be determined. If, for example, the creature's Energy level (E) reaches a value of E < 1, the creature becomes hungry and wants to eat. On the other hand, if the Energy level has risen to E > 1, the creature wants to mate with other creatures. Figure 74 shows these relationships between energy levels and feeding and mating behaviours.

Feeding: if E < 1 .... creature wants to eat text characters
it eats the same characters as its genetic code
("John" creature eats: "J', "o", "h", "n")
Mating: E > 1 .... creature wants to mate, if successful,
parents will exchange their genetic code
-> a child creature can be born
Evolution: Selection on faster creatures, as they can eat and
mate more frequently

Fig. 74 Creature's interaction parameters.

8.3.3.3 – Feeding

A creature whose Energy level has risen to E < 1 becomes virtually hungry and desires to eat text characters provided by the user through the system's GUI text input editor. The kind of text characters released depends solely on the user's decision. Creatures also have preferences for certain types of food and only eat text characters contained in their genetic message. For example, a creature whose genetic code is 'John' will only eat 'J', 'o', 'h', and 'n' characters. By eating text characters, the creature will manage to accumulate a certain amount of energy, and eventually its Energy level can again rise to E > 1. However, it might be necessary for the creature to eat several text characters as the creatures vigorously move while looking for text characters.

8.3.3.4 – Mating

Given that a creature has succeeded in obtaining energy at an amount E > 1, it will become a potential mating partner. It will now look for a suitable mate, whose energy level also lies above 1. The two potential parent creatures now will move toward each other and try to collide. If successful, the two parents exchange their genetic code through a cross-over operation and, as a result, a child creature can be born. This offspring creature carries the genetic code of its parents with some mutations. Figure 75 shows an example of a mating process.

Example of Mating Process

Parent creature (P1) and (P2), child (C)
| .... area of cross-over
^ ... location of mutation

Parent Creatures P1 and P2

```
(P1) Hello my name is Jenny.
           |        |        ^
(P2)  How are you Frank ?
           |       |
```

Child Creature C

(C) Hello are you is Jenny !

Fig. 75 Mating process and birth of child creature.

8.3.3.5 - Growth and Death

A creature's lifetime is not pre-determined but influenced by how much a creature eats. Through eating the creature increases its body size up to a maximum size of about four times the original body size. On the other hand, a creature will starve when it does not eat enough text characters and ultimately die and sink to the ground.

8.4 – Interaction and Evolution in Life Spacies II

In "Life Spacies" and "Life Spacies II", the constant movements, feeding, mating, and reproduction of the creatures result in a complex system of interactions that displays features of artificial evolution where selection favours faster creatures. Even though the selection pressure for the creatures is on catching the food fast and thus mating more frequently, users can reverse this process by "helping" slower creatures to gain enough energy and to mate as well. The users' decisions on how to write the text messages and on how and where to feed the creatures adds constant change and creates a system that features complex interactions between creatures as well as between users and creatures. The creatures' behaviours and survival are thus based on their genetic code, the users' interactions with them and the interactions between themselves. Figure 76 shows a screenshot of different creatures as they mate and feed on text characters, and Figure 77 shows the "Life Spacies II" life cycle with birth, feeding, motion, mating, reproduction and death.

Fig. 76 Complex behaviour among "Life Spacies II" creatures.

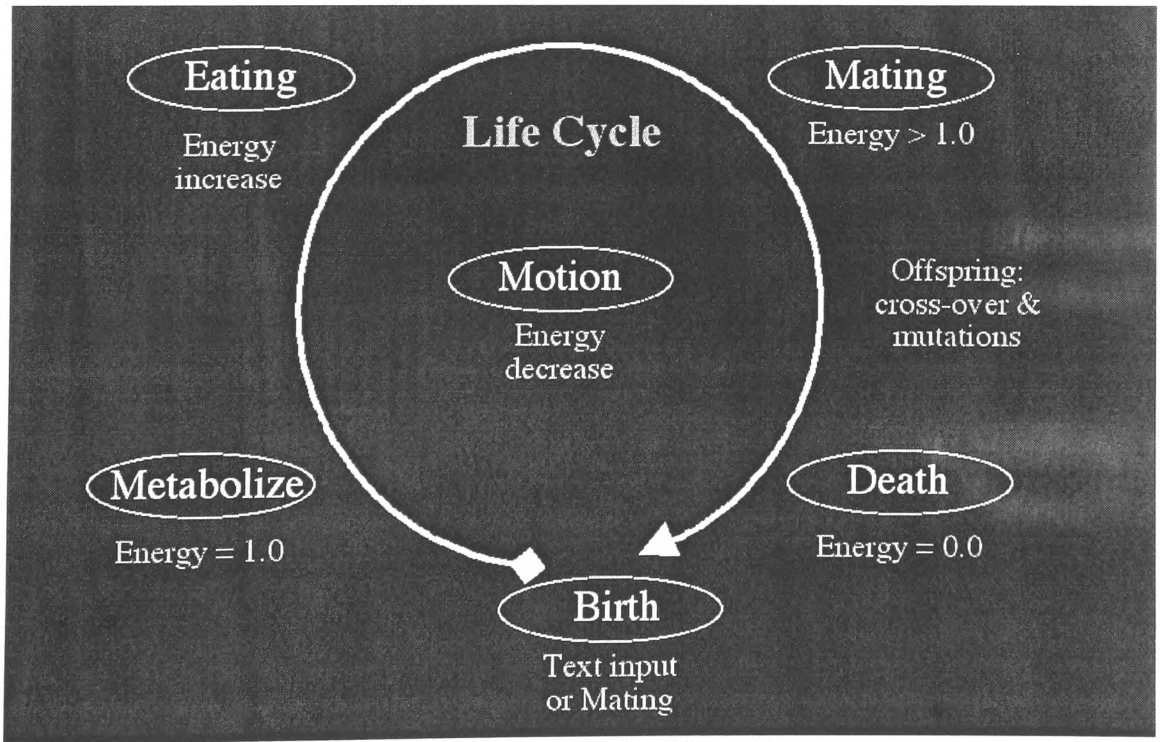


Fig. 77 "Life Spacies II" life cycle.

8.5 – Complexity Evaluation of Life Spacies II

As a result of the creatures' constant internal interactions, such as feeding and mating, and the users' interactions such as creating, protecting and supporting creatures, a system is created that features complex interactions between creatures as well as between users and creatures. When we go back to the definitions of Complex Systems (Section 2.2) and Complex Adaptive Systems (Section 4.1.), we can see that "Life Spacies II" displays the following general features associated with Complex Systems: to couple to each other, to adapt and organize, to mutate and evolve, to expand their diversity, to react to their neighbours and to external control, to explore their options, and to replicate. Out of the 11 required features, "Life Spacies II" satisfies 9 complexity requirements; only the learning function and the organisation of hierarchies of higher-ordered structures are not met. This is understandable when one considers that our system is designed for real-time practical use by exhibition visitors: implementation of a learning algorithm would have excessively slowed down the system and the real-time interaction between the creatures. In the future, with increased calculation speed, it will certainly become possible to include a learning algorithm, which should help the creatures in "Life Spacies II" to organize hierarchies of higher-ordered structures within the system as well. Detailed descriptions of "Life Spacies II" are also provided in (Sommerer *et al.*, 1999b, 1999c, 1999d, 1999e, 1999f, 2000a, 2000b, 2000c, 2000d, 2001b) and an interactive demonstration of the "Life Spacies II" software (on CDRom) has been attached and submitted together with this thesis.

9 - Experiments in Modeling and Generating Complexity for Interactive Art on the Internet

9.1 - Introduction

The Internet is an ever-expanding database of images, text and sound files, currently containing several billion documents. These data and their internal organization are constantly changing, as new documents are being uploaded, new web sites are being created, and old links are being deleted. New connections between various sites are also constantly built, and the Internet itself has basically become an evolving, re-connecting and reconfiguring network of user-driven data input and output.

After having successfully modeled a CAS for interactive art (see "Life Spacies II" described in Chapter 8), I became interested in exploring other forms of complexity by using the Internet as a platform for inquiry and experimentation. Since 1999, I have created various interactive artworks for the Internet, and in the following sections I will describe these systems in more details. As mentioned earlier, these systems are not directly intended as CAS or complexity models but instead function as experiments to model aspects of interactive complexity on the Internet.

9.2 - VERBARUM - Modeling Complexity for Interactive Art on the Internet

9.2.1 - Introduction

In 1999, I developed my first prototype toward the objective of applying principles of Complex Systems to the creation of interactive artworks on the Internet. This system is called VERBARIUM, and it is an interactive web site where users can choose to write email messages that are immediately translated into visual 3-D shapes. As the on-line users write various messages in the Graphical User Interface (GUI) of the

VERBARIUM web site, these messages are translated by our in-house "text-to-form editor" into various 3-D shapes. By accumulation, these shapes can collectively create more complex image structures than the initial input elements. Increased users interaction with the system is expected to cause increasingly complex image structures to emerge over time.

9.2.2 - System Description

VERBARIUM is available on-line at the following web page: http://www.fondation.cartier.fr/verbarium.html. It was created for the collection of the Cartier Foundation in Paris (Sommerer and Mignonneau, 1999g). Detailed information on this system was also published in (Sommerer and Mignonneau, 1999f; 2000a; 2000d; 2002a; 2002b). The on-line user of VERBARIUM can create 3-D shapes in real time by writing a text message within the interactive text input editor in the lower-left window of the web site. Within seconds the server receives this message and translates it into a 3-D shape that appears in the upper-left window of the web site. Additionally, this shape is integrated into the upper-right window of the site, where all messages transformed into shapes are stored in a collective image. An example screenshot of the VERBARIUM web site is shown in Figure 78.

VERBARIUM consists of the following elements:

1. a JAVA based web site
2. an interactive text input editor (lower-left window in Fig. 78)
3. a graphical display window to display the 3-D forms (upper-left window in Fig. 78)
4. a collective display window to display all of the produced 3-D forms (upper-right window in Fig. 78)
5. a genetic Text-to-Form editor to translate text characters into design functions

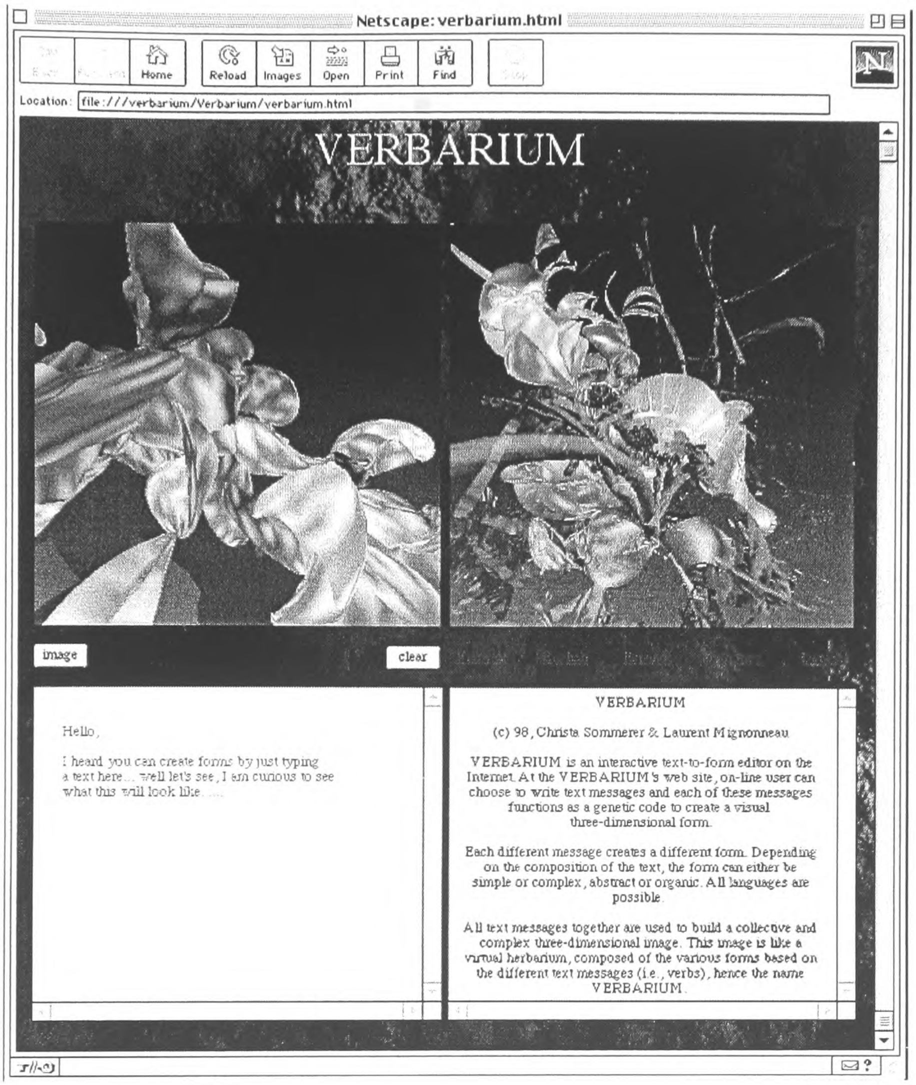


Fig. 78 VERBARIUM web page.

9.2.3 - Text-to-Form Editor

We have set up a system that uses the simplest possible components for a 3D form that can subsequently model and assemble more complex structures. The simplest possible form we constructed is a ring composed of 8 vertices. This ring can be extruded in x, y and z axes, and during the extrusion process the ring's vertices can be modified in x, y and z axes as well. Through addition and constant modification of the ring parameters, the entire structure can grow, branch and develop. Different possible manipulations, such as scaling, translating, stretching, rotating and branching of the ring and segment parameters creates diverse and constantly growing structures, such as those shown in Fig. 79.

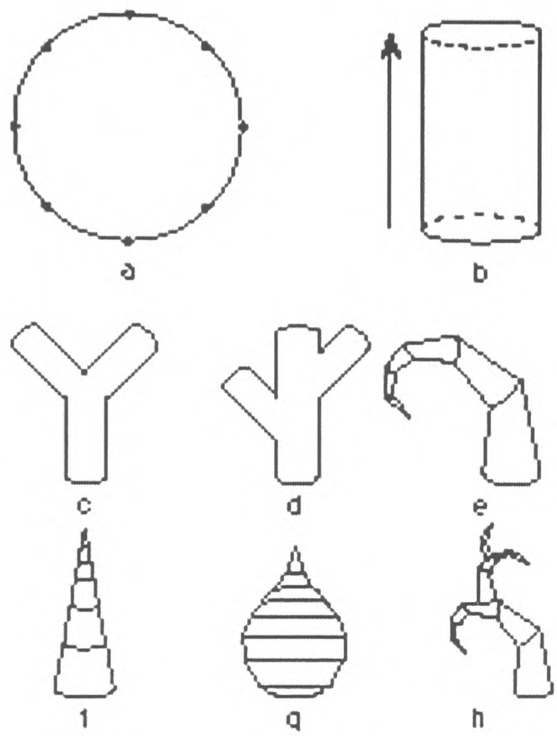


Fig. 79 Example of VERBARIUM's growing structures.

Figure 79 a shows the basic ring with eight vertices, and Fig. 79 b shows the extruded ring that forms a segment. Figures 79 c and 79 d show branching possibilities, with branching taking place on the same place (internodium) (79 c) or on different internodiums (79 d). There can be several branches attached to one internodium. Figure 79 e shows an example of segment rotation, and Fig. 79 h shows a combination of rotation and branching. Figures 79 f and 79 g are different examples of scaling. In total, there are about 50 different design functions, which are organized into the design function look-up table (Fig. 80). These functions are responsible for "sculpting" the default ring by modifying its vertex parameters.

function1 translate ring for certain amount (a) in x

function2 translate ring for certain amount (a) in y

function3 translate ring for certain amount (a) in z

function4 rotate ring for certain amount (b) in x

function5 rotate ring for certain amount (b) in y

function6 rotate ring for certain amount (b) in z

function7 scale ring for certain amount (c) in x

function8 scale ring for certain amount (c) in y

function9 scale ring for certain amount (c) in z

function10 copy whole segment(s)

function11 compose a new texture for segment(s)

function12 copy texture of segment(s)

function13 change parameters of RED in segment(s) texture

function14 change parameters of GREEN in segment(s) texture

function15 change parameters of BLUE in segment(s) texture

function16 change patterns of segment(s) texture

function17 exchange positions of segments

function18 add segment vertices

function19 divide segment in x to create branch

function20 divide segment in y to create branch

function21 divide segment in z to create branch

function22 create new internodium(s) for branch(es)

function23 add or replace some of the above functions

function24 randomize the next parameters

function25 copy parts of the previous operation

function26 add the new parameter to previous parameter

function27 ignore the current parameter

function28 ignore the next parameter

function29 replace the previous parameter by new parameter

....

function50

Fig. 80 VERBARIUM's design function table.

The translation of the actual text characters of the user's email message into design function values is done by assigning ASCII values to each text character according to the standard ASCII table, as shown in Figure 60 in Section 8.2.4. The actual encoding mechanisms between the actual ASCII values of the text message and the assignment between random functions and design functions follows the same principle I designed for the "Life Spacies II" system. This mechanism was explained in-depth in Section 8.2.4 and was shown in Figure 61.

The basic "module" in the VERBARIUM system is a ring which can develop and assemble into segments. It can so grow and branch to create more complex structures as the incoming text messages modify and "sculpt" this basic module by the available design functions, as shown in Figure 80.

9.2.4 - User Interaction

Depending on the complexity of the incoming text messages, the 3-D forms become increasingly shaped, modulated and varied. As there is usually great variation among the texts, the forms themselves also vary greatly in appearance. As a result, each individual text message creates a very specific three-dimensional structure that can at times look like an organic tree or at other times look more like an abstract form. All forms together build a collective image displayed in the upper-right window of the web site; it is proposed that the complex image structure that emerges represents a new type of structure that is not solely an accumulation of its parts but instead represents the amount and type of interactions of the users with the system. Another example of forms created by a different text message is shown in Figure 81, this time with text written in French.

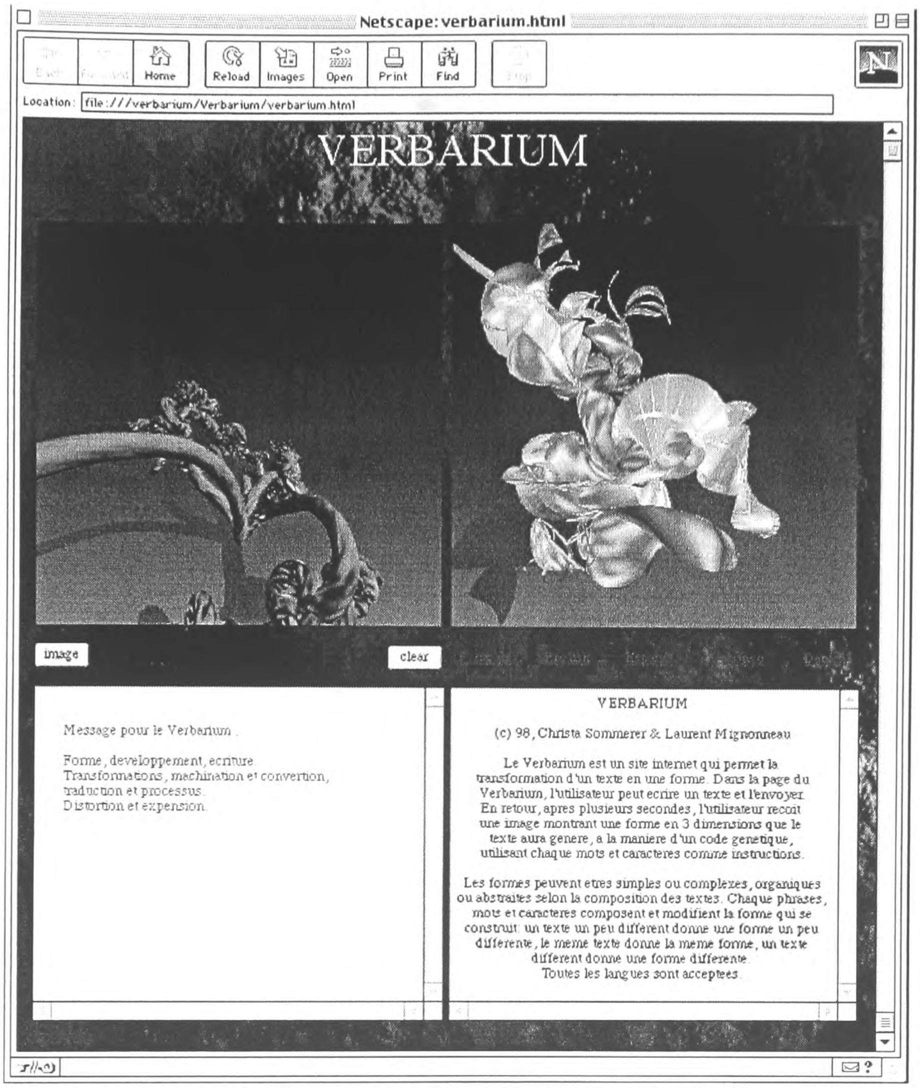


Fig. 81 VERBARIUM Web site - with French text input.

9.2.5 – Summary and Complexity Evaluation

VERBARIUM enables on-line users to create 3D shapes by sending text messages to the GUI of the web site. Using our text-to-form editor, this system translates the text parameters into design parameters for the creation and modulation of 3D shapes. These shapes can become increasingly complex as the users interact with the system. A collective image hosts and integrates all of the incoming messages that have been transformed, and as users increasingly interact with the system an increasingly complex collective image structure emerges. Figures 82 and 83 show how the collective image has become increasingly complex through user interaction. As it is no longer possible to deconstruct the collective image into its initial parts, some of the features of complex systems, such as variety and dependency (as described in Sections 2.4.1 and 2.4.2), irreducibility (as described in Section 2.4.3.), the ability to surprise (as described in Section 2.4.7), symmetry-breaking (as described in Section 2.4.9), and the notion that the whole is more than the sum of its parts (as described in Section 2.4.12) are thought to have emerged.

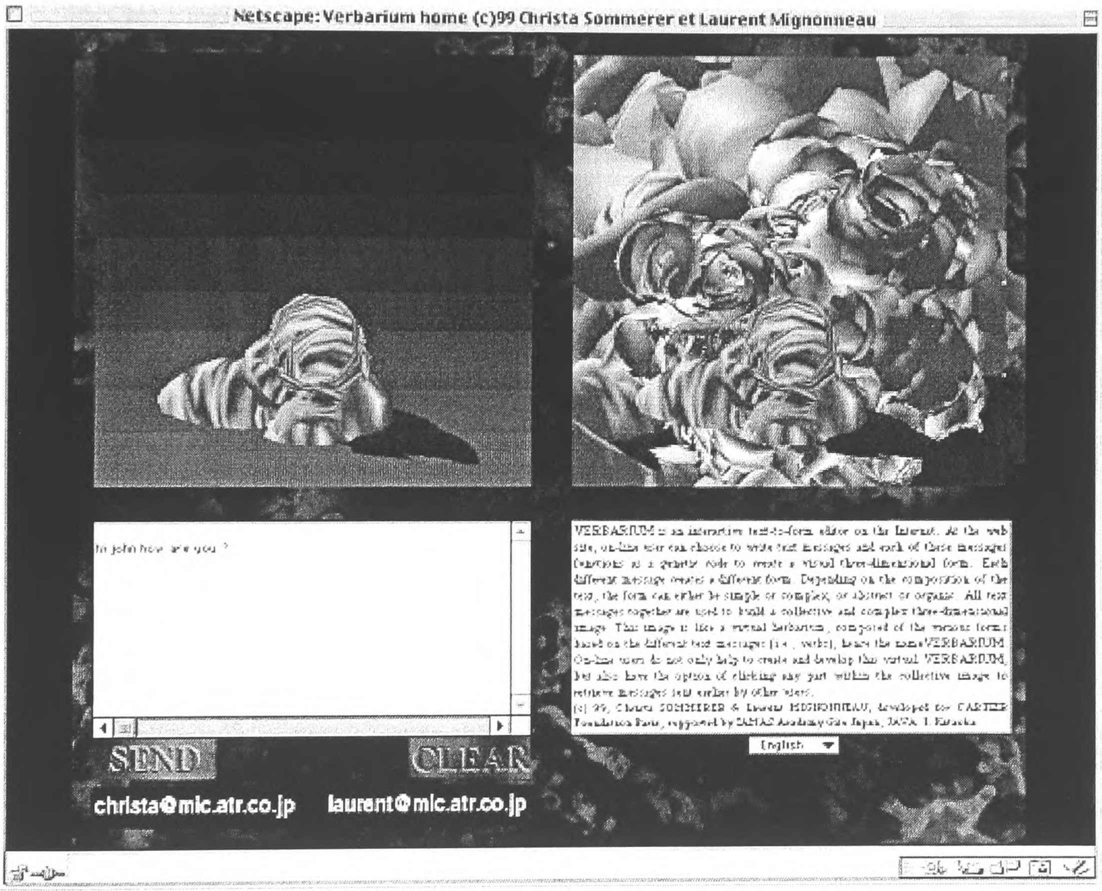


Fig. 82 VERBARIUM Web site - with simpler text message.

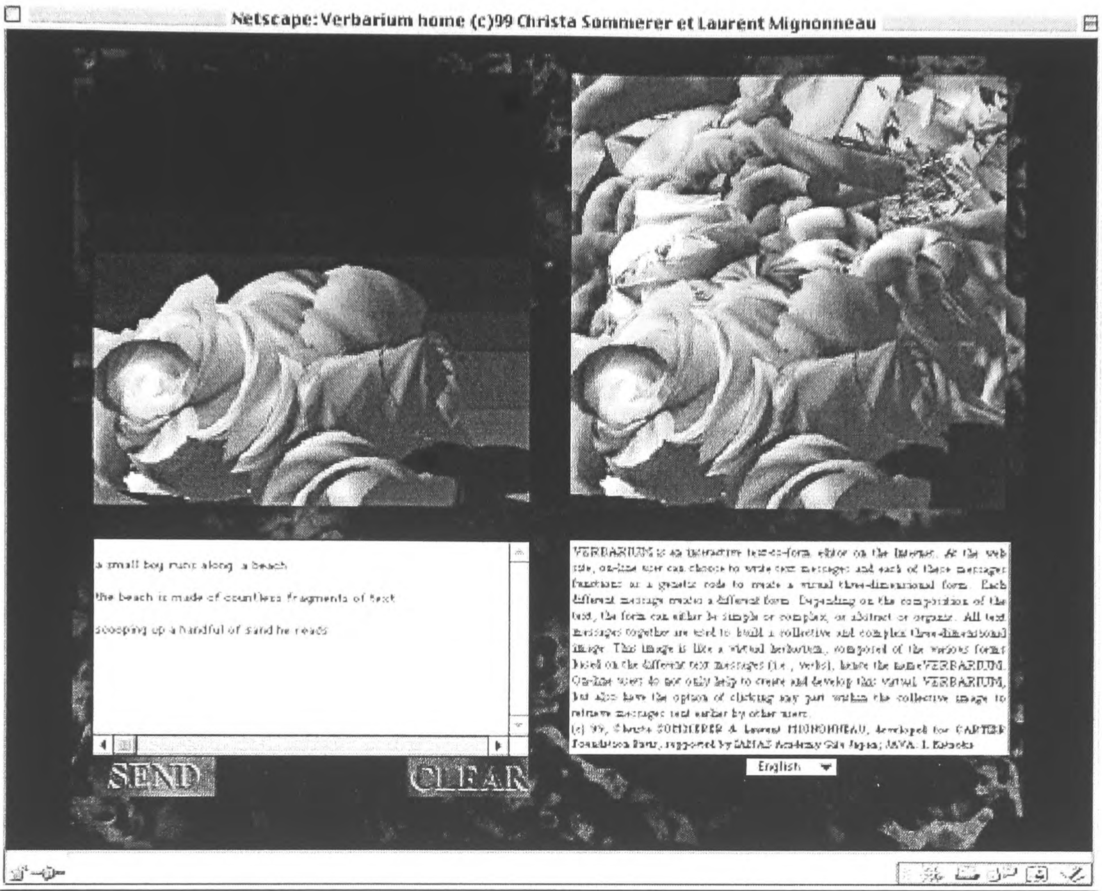


Fig. 83 VERBARIUM Web site - with more complex text message and a more complex collective image.

9.3 - Multi-Modal Interaction with Complex Data on the Internet: Riding the Net, The Living Room and The Living Web

9.3.1 - Introduction

When browsing the Internet we are nowadays confronted with the standard solutions of how to display text, image and sound data in the conventional Internet browsers such as Netscape (2001) or Internet Explorer (2001). While information in these browsers is certainly organized very efficiently, it can sometimes be cumbersome and tiresome to retrieve information or to just browse and enjoy the visual stimuli of the vast amounts of text, images and sounds available.

Aiming to create systems that can model aspects of complexity on-line, I have become interested in tapping into the existing complexity of available data instead of designing complex systems from the bottom up. I created three on-line experimental systems that detect the users' speech input to generate keywords which in turn generate downloads of corresponding image, text and sound data. Users of the systems can interact with these data through intuitive and multi-modal interfaces, creating interaction experiences that allow them to experience the complexity of the Internet depending on the complexity of their own input data. Three of the systems I created since 1999/2000 will be described in the following sections.

9.3.2 - Knowledge Discovery

The last decade has seen an explosive growth in the generation and collection of data. Advances in data collection, widespread use of bar codes for most commercial products, and the computerization of many business and government transactions have flooded us with data and generated an urgent need for new techniques and tools that can intelligently and automatically assist in transforming this data into useful knowledge. Data mining (Han and Kamber, 2000), knowledge discovery (Witten and Frank, 1999), and information retrieval (Salton, 1983) are areas of research that deal with issues of how large amounts of data on the Internet can be organized and retrieved efficiently.

9.3.3 - Riding the Net

In my aim to create an image browser that is more entertaining, playful and intuitive to use, and which is able to browse through complex data spaces, I developed an interactive system called "Riding the Net" (Mignonneau *et al.*, 2001a and 2001b). Here users can use speech communication to retrieve images from the Internet, watch these images as they stream by, and interact with them by touching them. Two users can interact in this system simultaneously, and as they communicate their conversation will be supported and visualized in real-time through images streamed from the Internet.

9.3.3.1 - System Description

The system consists of an interactive window where two users, who sit opposite of each other, can communicate through speech. Microphones attached to the users pick up keywords of their conversation and feed them into the system's speech recognition engine. These keywords are then used by the image retrieval server to search and download their corresponding images from the Internet. The images themselves are then handled by the graphics manager and streamed in 3D onto a large interactive window screen. Here users can watch the different image icons simultaneously as they move, touch them with their hands, stop them and retrieve their corresponding URLs. Figure 84 shows the system overview and the link between speech recognition system, image retrieval server, graphics manager, and touch screen interface.

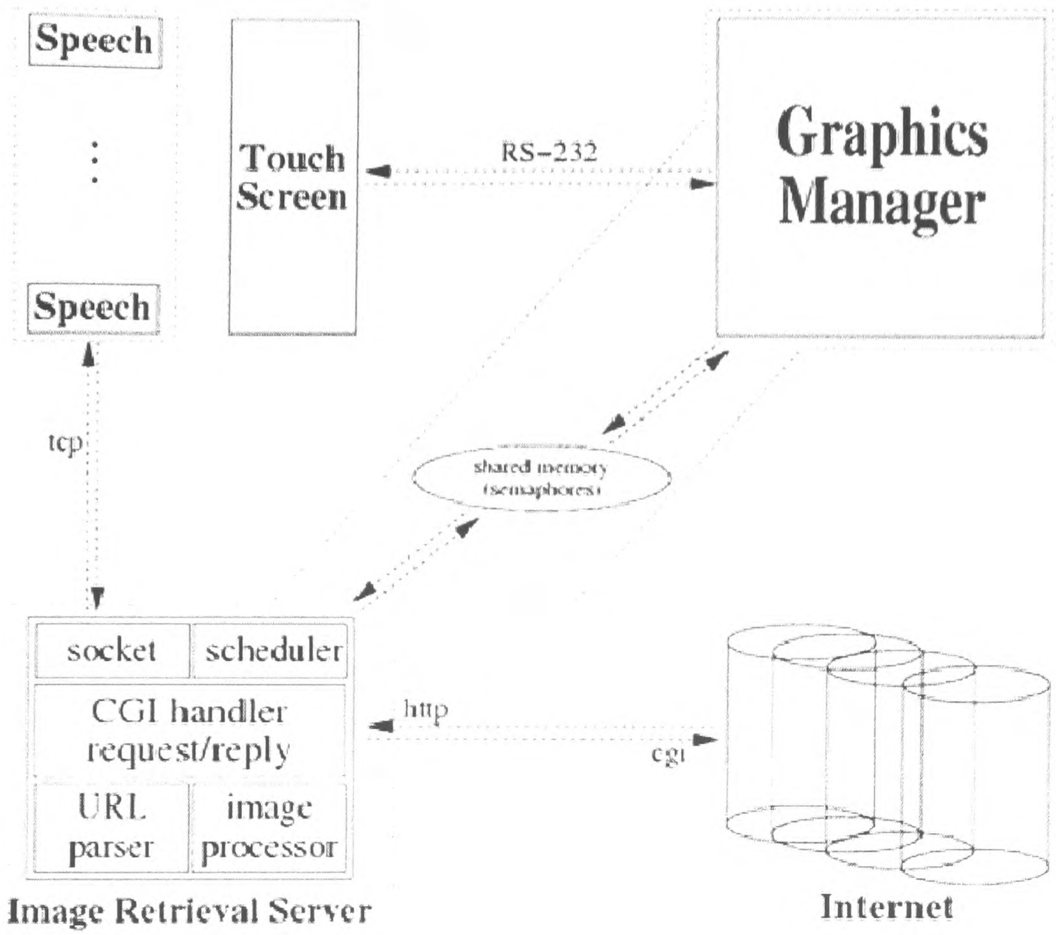


Fig. 84 “Riding the Net” System Overview.

9.3.3.2 - Speech-based Image Retrieval

We use off-the-shelf speech recognition software (IBM Via Voice) to detect keywords in the users' conversations. These keywords are then fed into the CGI handler. This component creates threads that send various requests to different Internet image servers, such as AltaVista and Google.

Once an URL has been parsed and the contained images on this page have been retrieved, we process them for further display by cropping unwanted frames, resizing them and attaching the image information (SHM). Figure 85 shows the image retrieval process based on the speech recognition input.

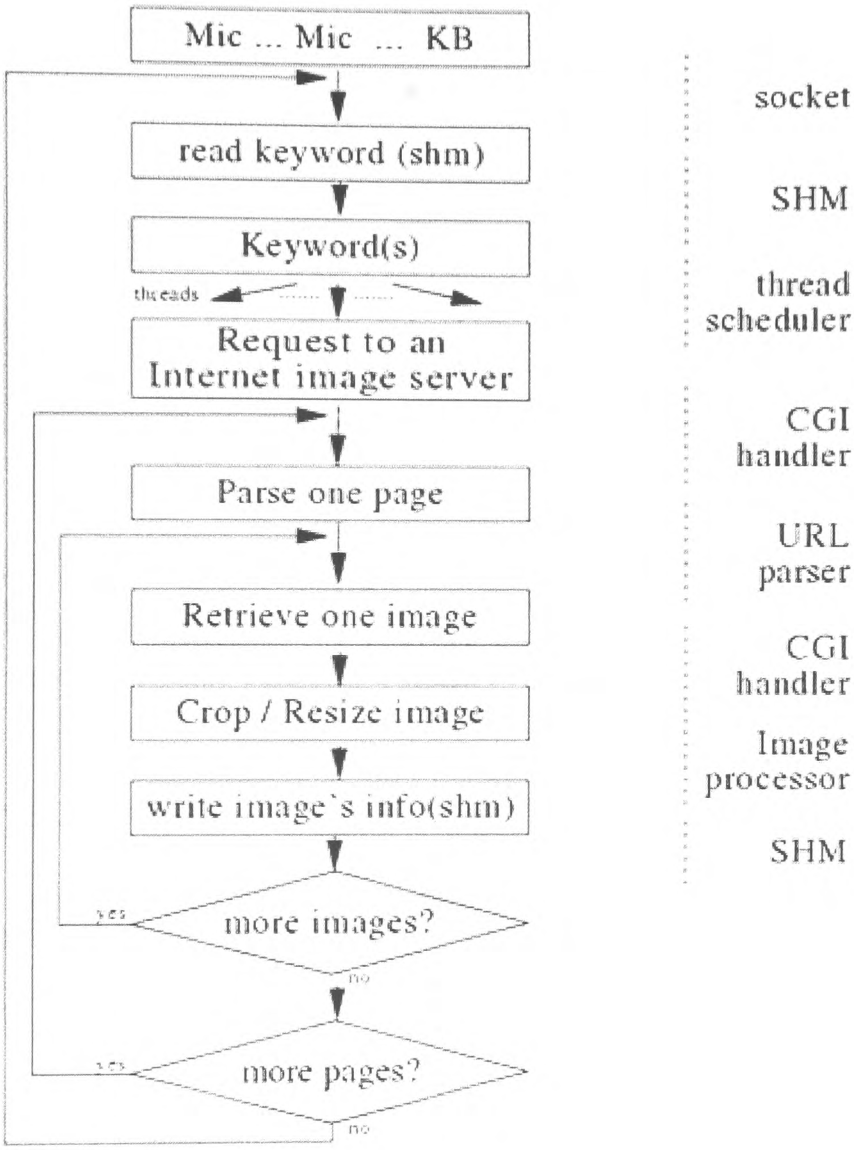
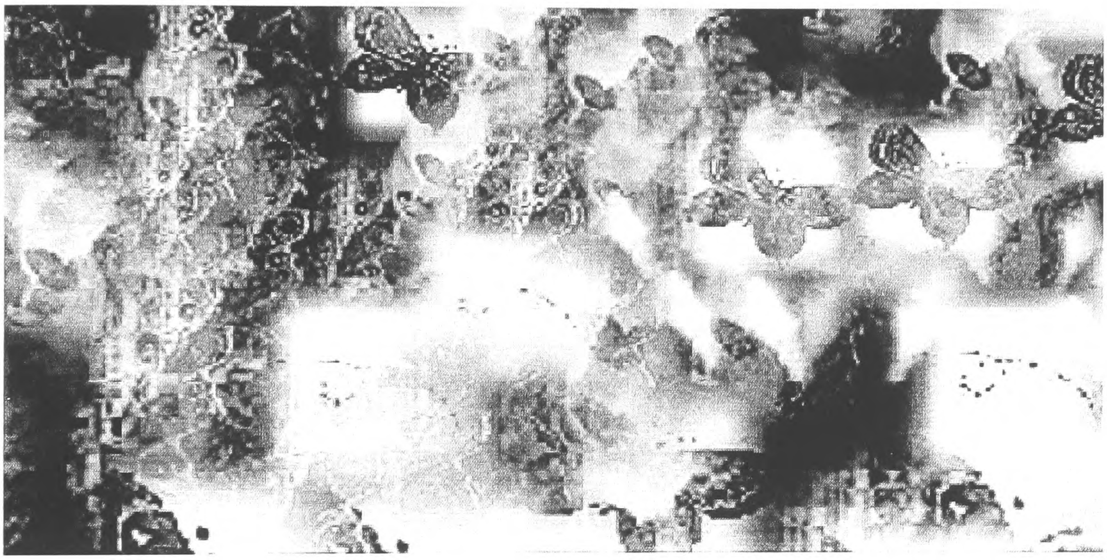


Fig. 85 Image retrieval process in "Riding the Net".

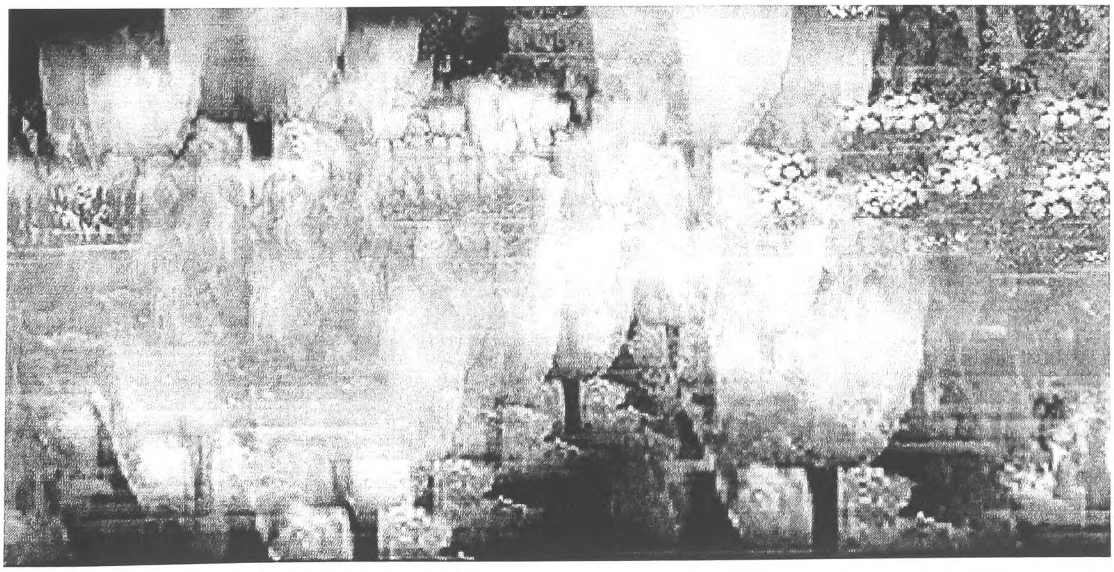
9.3.3.3 - Real-time Image Streaming

The image icons derived from the speech recognition and image retrieval process are then collectively displayed in 3D on the system's interactive window screen. Here image icons are streamed from the respective view of each user, arriving from either the left or right side of the screen. All images stream toward each other before they leave the screen and are replaced by new images derived from new keywords spoken by the two users.

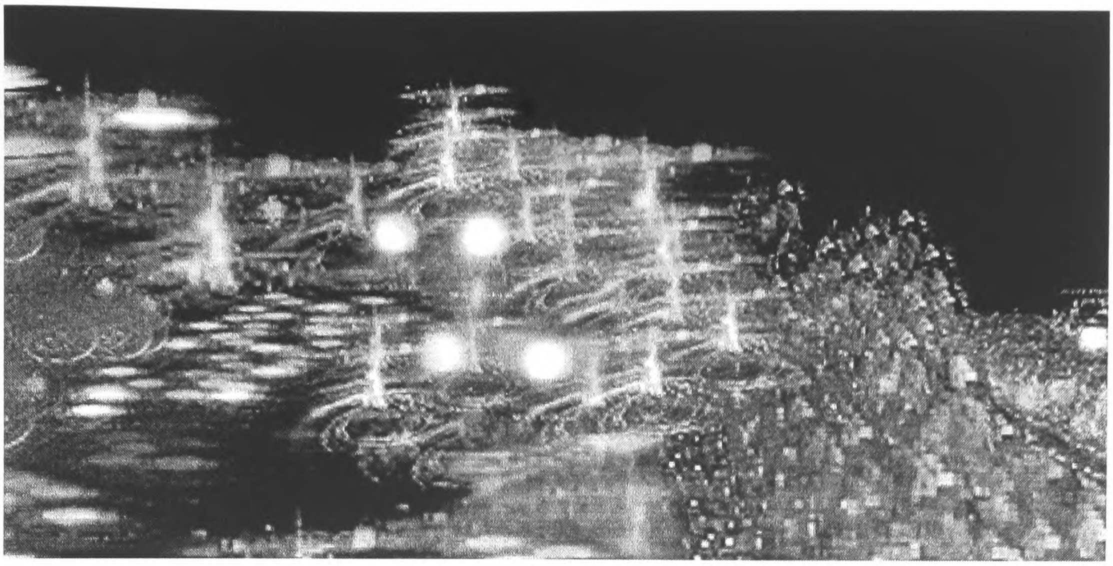
In addition to seeing the image icons appear on the screen, users can also see what the speech recognition engine has detected; a small text area inside the interactive window display shows the detected words. This provides the users with some feedback on the accuracy of the speech recognition system and the images that are going to appear. When users for example speak about "galaxy" or "flowers", these keywords would be detected and different images of galaxy or flowers would be downloaded from the Internet. The overall image scenario on the window surface thus constantly changes, since it is a direct interpretation of the users' dialogue. Examples of image scenarios and their corresponding keywords are shown in Fig. 86 a-c.



(a)



(b)



(c)

Fig. 86 a-c Screenshots of image scenarios derived from various keyword inputs: butterfly (a), flower (b), galaxy (c).

9.3.3.4 - Touch-based Interaction

Users can not only generate new images through their conversations but also touch these image icons on the interactive window display. Touching an image icon temporarily halts the image, and users can look at this specific image icon in more detail. When they do this, the exact URLs for this specific image icon can also be provided and downloaded onto a separate computer screen. This allows the users to find out where this image came from by following the URL.

The hand detection on the interactive window display is done through an in-house interface technology based on a grid of infrared (IR) emitters and detectors. When the user puts his hand onto the window's surface, the infrared detection at that particular location is interrupted. The information about the interruption's x and y value can then be sent to the serial interface, which relays these data to the interface protocol. After A/D conversion these date are then linked to the exact location of the image icon so that the users can stop specific icons under their hands.

The overall size of the window display is 150 cm x 110 cm, and around 68 X-emitters and detectors as well as 46 Y-emitters and detectors are used. This provides sufficient precision for hand detection since an image icon and a typical user's hand usually cover around 8 X and 6 Y emitters and detectors.

The system also allows for multiple-hand detection, so two or more users can touch and halt image icons on the screen simultaneously. Figure 87 shows the hand detection system and the grid of (invisible) IR sensors.

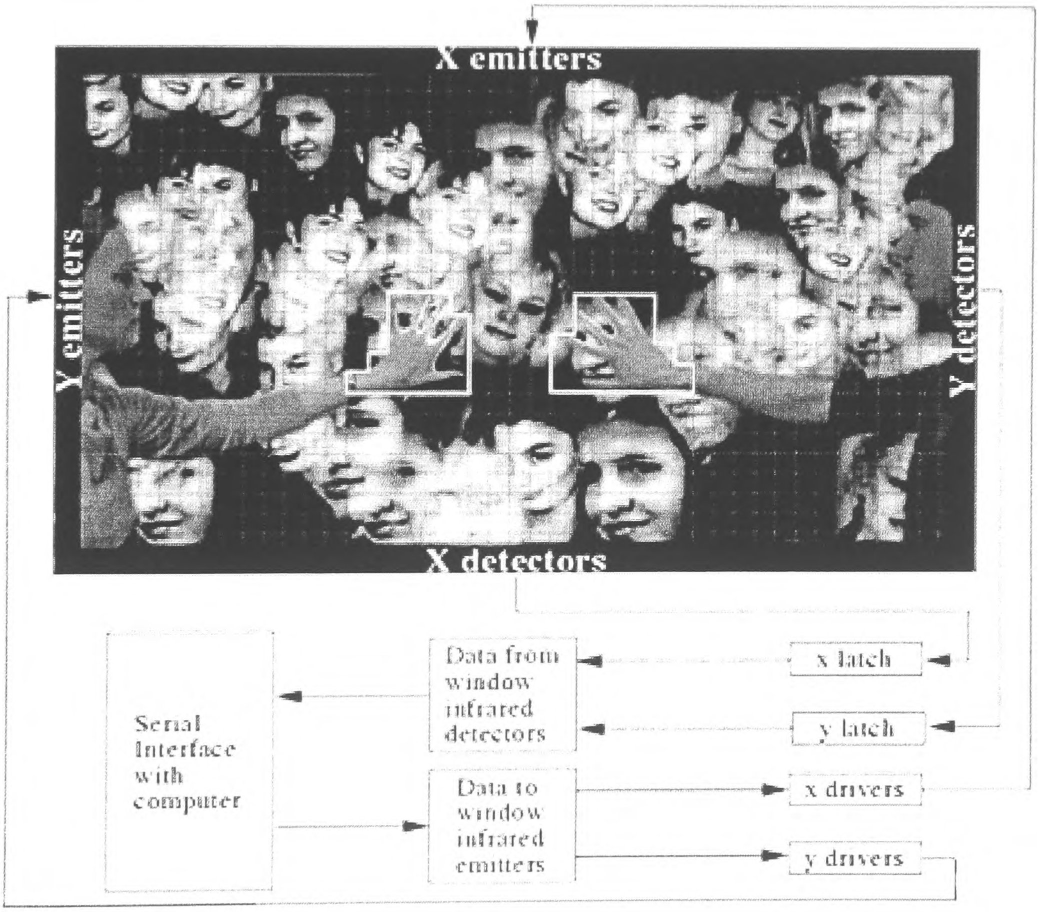


Fig. 87 Multi-User Touch Screen Interface.

9.3.3.5 - Multi-modal Interaction and Communication

Users communicate with each other while sitting on chairs in front of the interactive window screen. Microphones pick up the user's conversation and use detected keywords to stream their corresponding images from the Internet. As users engage in communication, their conversation becomes visualized and the appearing images can themselves be used as subsequent communication stimuli. As there is never an exact correspondence between the spoken words, the detected words, and the downloaded images, users will find many surprises, unexpected images and words. They can use this for engaging in an entertaining journey through the large amounts of data available on the Internet. Furthermore, by being able to touch the image icons on the screen, users can engage in multimodal interaction and communication experiences that include the users' visual, audio and touch senses. Figures 88 and 89 show users as they communicate with each other in front of the interactive window screen and as they touch image icons on the screen.

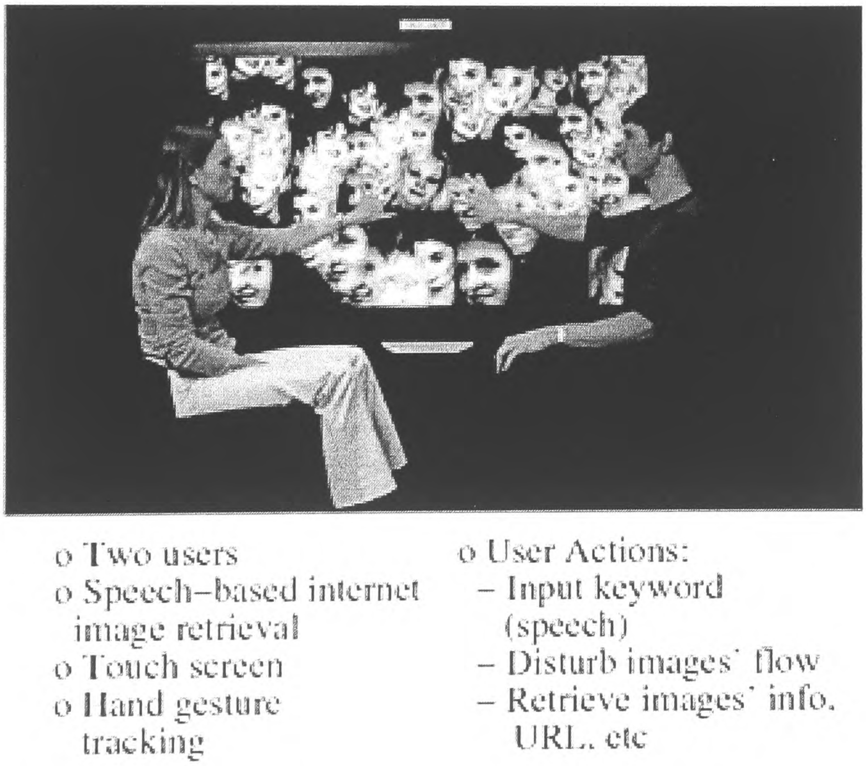


Fig. 88 User Interaction and Communication in the "Riding the Net" system.

Fig. 89 Users interacting with the "Riding the Net" system through speech and touch input.

9.3.4 - The Living Room - Web-based Image and Sound Environment

In 2001, I adapted the "Riding the Net" image retrieval software for an interactive information environment, called "The Living Room". This system consists of four interactive screens as shown in Figure 90. It was developed for the "Bo01-Living in the Future" architecture exhibition held in Malmoe, Sweden in May 2001 (Sommerer, Mignonneau and Lopez-Gulliver, 2001b).

Fig. 90 "The Living Room" setup with four interactive screens that display information streamed from the Internet.

Users in this system enter a 6 x 6 meters large space that consists of four 4 x 3 meters screens and as they talk, microphones placed on the ceiling of the space detect their conversations. Detected keywords are then used to generate word icons, which start to appear and float on the four screens. Users can touch any of these word icons and their touch will trigger the downloading of corresponding images from the Internet, as shown in Figure 91.

Fig. 91 "The Living Room" - a user touches word icons to download their corresponding images and sounds from the Internet.

A custom-designed camera detection system uses infrared light to allow multiple-user hand detection on the 4 x 3 meters projection screens. When users for example touch the word icon "lemon", images of lemons are downloaded from the Internet, appearing as constant radial image streams as long as the user chooses to grasp this keyword with his hand. In addition to downloading images, corresponding sound files are streamed as well. In the case of the example "lemon", our system searches for sound files with the "lemon.mp3" tag or sound files that contain "lemon" in their title or in the composer's or performer's name.

In "The Living Room", ten different search engines are being called with up to 30 simultaneous requests. As users speak, a constant stream of new keywords is being generated, which in turn generates a constant stream of new word icons. Up to 30 users in the system can choose to touch the various word icons, which will generate constantly changing image and sound downloads from the Internet. As a result of these multi-user interactions, a dynamic, self-organizing, and constantly changing information space emerges. It represents the users' individual conversations, their individual interests in certain topics, and their collective interaction with the shared information. As in "The Living Room", both the imprecision of the speech recognition system and the randomized choice of images from the various search results are used intentionally to create a dynamic system that is unpredictable, full of surprise, and compliant with some of the definitions of a complex system. Figure 92 shows two users as they interact with "The Living Room" and share some of the downloaded image and sound information.

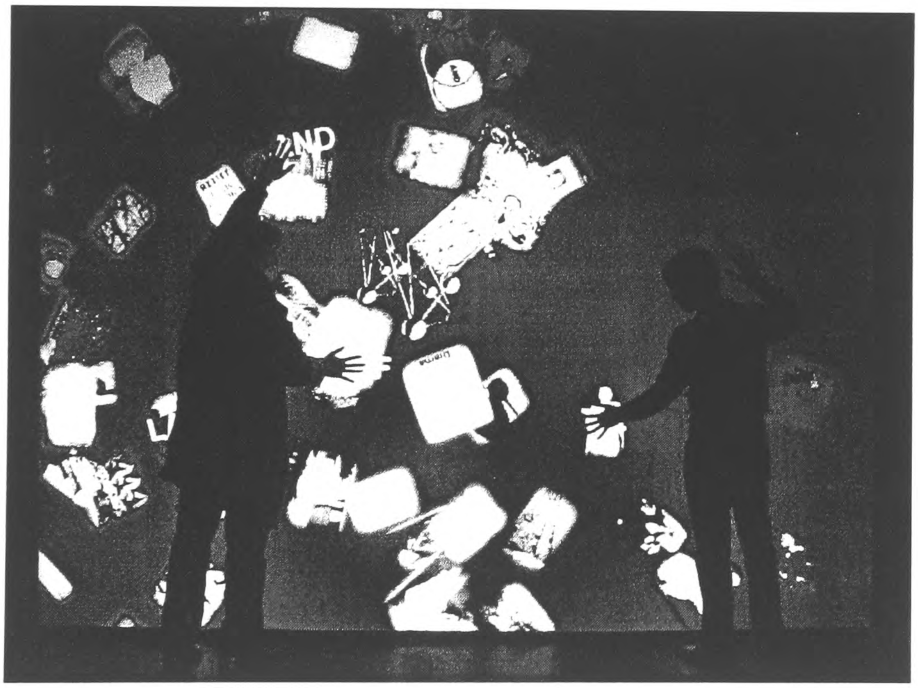


Fig. 92 Multiple users can simultaneously interact in "The Living Room" system.

9.3.5 - The Living Web - An Immersive Web Environment

In May 2002, I adapted "The Living Room" software to the 3D immersive environment of the CAVE™ system. The CAVE™ was invented in 1991 by DeFanti and Sandin at the Electronic Visualization Laboratory at the University of Chicago

(Cruz-Neira, Sandin and DeFanti, 1993). It is a surround-screen virtual environment that consists of four to six large screens that make up a room where computer-generated images can be projected onto the screens in stereo. The user in this system wears a pair of lightweight shutter glasses for stereo viewing. In addition, the user can also typically operate a wand interface that lets him control the virtual objects displayed in stereo on the screens. The feeling of 3D immersion is very convincing as the user can move freely and the images are displayed seamlessly on all four to six screens.

Having already created a four-sided projection space in "The Living Room", I became interested in creating a more immersive environment for the Internet, where users can actually "enter the Internet" and interact with the available image and sound information in three dimensions. Adapting our software to the CAVE™-specific AVANGO libraries (which calculate stereo vision and multi-channel performance), we created "The Living Web" software. The system has been shown since June 1, 2002 at the "Art-of-Immersion Festival" in Bonn, Germany (Sommerer, Mignonneau and Lopez-Gulliver, 2002). Here users can physically immerse themselves into the data space of the "The Living Web" and interact with image data and sound data through a specifically designed tweezers interface, as shown in Figure 93. When users talk into their headset microphones, images that relate to their conversations are streamed from the Internet and displayed in 3D in such a way as to surround them. By grabbing one of the floating images, the user can retrieve more information about this specific image (for example its URL), place the icon in a 3D space for bookmarking it, and sort the various selected icons as 3D bookmarks to create further links, weights of interests, and connections between the various selected topics.

Fig. 93 A user interacts with the "The Living Web" in the CAVE™ environment, using a specifically designed tweezers interface that allows her to grab image data, place them in a 3-D space, sort them, and bookmark them.

9.3.6 - Complexity Evaluation of Riding the Net, The Living Room and The Living Web

In the previous sections I have introduced three Internet-based systems that are designed for interacting with complex image, text and sound data from the Internet in an intuitive, novel and entertaining fashion. Users in these systems become intensively engaged in the creation of data through their speech input, which in return triggers the downloading of image and sound data. While users have some control over what kind of image and sound downloads are triggered, the sheer quantity of available information makes straightforward selection impossible. For each keyword, typically several hundred or at times several thousand image and sound documents are available and users can typically only perceive a fraction of the available data. To manage this complex and constantly changing database of images and sounds and to allow intuitive as well as creative data browsing, these systems were designed to deal with randomness and order, allowing partly directed and partly undirected searches.

While these systems are not directly intended as a data mining applications in the traditional sense of accessing data in a straightforward manner, they can be useful tools for visual and intuitive browsing through large and complex amounts of image and sound data. In these systems, up to several hundred image icons can be displayed simultaneously. As these image icons keep streaming toward the users, they can discover new and unexpected images, watch their URLs, create bookmarks and use those to further create links between related information. Especially for users who do not want to use the traditional computer screen, mouse and keyboard, our systems provide easy access with unencumbered and intuitive interfaces (Mignonneau, Sommerer and Lopez-Gulliver, 2001a and 2001b).

10 - Conclusions and Future Work

10.1 - Comparison of Initial Theory with the Results

The objective of my thesis was to construct artistic and interactive systems that apply principles of Complex Adaptive Systems (CAS) and Complexity to interactive art. My aim was to construct artworks that could increase their internal complexity by linking user interaction data to the system's internal software structures. In Chapters 8 and 9, I introduced various interactive art works, and now I would like to analyze how and whether these system did in fact fulfil some of the CAS requirements or the general complexity measures. Let us now look at our findings and analyse these results.

10.1.1 - Life Spacies II Complexity Evaluation

When we analyse the results obtained through our "Life Spacies II" system, we can see that the design of the genetic code for the artificial creature was directly linked to the user's input data. The genetic code of the creature is defined by how the user writes a text message, and the genetic code in turn decides the creature's overall ability to move. Movement is an expression of the creature's fitness, and this determines its survival and possible evolution. As we have seen in the definitions of complexity and Complex Adaptive Systems (CAS), evolution is often seen as a mechanism for increasing complexity. In "Life Spacies II" the second significant parameter for a creature's behaviour, and ultimately its survival and evolution, is its energy value. This value is also linked to the user's interaction, specifically how frequently, where, and with which food (text) a user decides to feed a creature. In other words, the entire "Life Spacies II" life cycle and balance of creation, metabolism, combat, reproduction, adaptation and evolution is a feedback loop between the users' input parameters and the creatures' own internal decision parameters. The constant update and feedback between these internal and external values thus determine a system's overall balance, evolution and increase in complexity.

While various scientific models of CAS have shown us how the interaction between artificial agents (or creatures) can enable an open-ended evolutionary structure with increased complexity (see the models by Dawkins (1986), Reynolds (1987), Ray (1991), Yaeger (1994), Holland (1994), and Langton *et al.* (1995)), artists and designers have created generative artworks, or entertainment software, that have often used a more predefined set of agents with limited design decisions by the users (see the models by Ventrella (1996, 1995), Grand *et al.* (1997), Spofford (1998), Heudin (1998), Virtual Fishtank (1998), Annunziato (2000), and Hurry *et al.* (2000)). "Life Spacies II", on the other hand, was designed as a system that tries to give maximum design flexibility and maximum interaction decision to the users, aiming to create an open-ended system that increases its overall complexity through user interaction.

When we go back to the definitions of Complex Systems (Section 2.2) and Complex Adaptive Systems (Section 4.1), we can say that the creatures in "Life Spacies II" fulfil the following complexity criteria: they couple to each other, they adapt and organize, they mutate and evolve, they expand their diversity, they react to their neighbours and to external control, they explore their options, and they replicate. We can thus conclude that "Life Spacies II" meets almost all essential criteria of a CAS, making it natural to consider this system a Complex Adaptive System.

10.1.2 - VERBARIUM Complexity Evaluation

With the on-line system VERBARIUM, I have introduced a mechanism that can encode text messages into 3-D growing shapes. The ASCII values of the on-line users' written text messages are used to modulate a ring algorithm which can grow in three dimensions, branch out, divide, and leave a 3-D trace in an on-line picture. Shapes can become increasingly complex as on-line users interact with the system. A collective image hosts and integrates all of the incoming messages that have been transformed into growing 3-D shapes, and as users increasingly interact with the system, an increasingly complex collective image structure emerges. While the system was not modeled as a CAS and does not feature reproduction, evolution, learning or adaptation, it seems to satisfy some of the other complexity definitions as outlined in Section 2.4. These features are variety, dependency, irreducibility, ability to surprise, symmetry-breaking, and the notion that the whole is more than the sum of its parts.

Other artists and designers (see the models by Kerne (2001), Wisniewski *et al.* (1999), Freude *et al.* (1999) and Karjalainen *et al.* (2000)) also increasingly use the Internet to model interactions between users or between users and on-line artificial characters. Even though these systems are not explicitly designed as complex systems, several complexity criteria are fulfilled in these systems as well. While they might not fully qualify as Complex Systems (as defined in Section 2.2), their inherent features do model properties often associated with complex systems. Therefore, these systems, including VERBARIUM, can become a good starting point for experimental research on how to model artistic on-line complex systems or on-line CAS in the future.

10.1.3 - Riding the Net, The Living Room and The Living Web Complexity Evaluation

While users in the VERBARIUM system create increasingly complex shapes by writing text messages into the GUI of this web site, users in the "Riding the Net", "The Living Room", and "The Living Web" systems interact with existing image and sound data on the Internet through multi-modal interaction. The users' speech input is used to generate keywords which in turn generate downloads of corresponding image and sound data. While users have some control over what kinds of images and sounds are triggered, the sheer quantity of available information makes straightforward selection impossible. For each keyword, typically several hundred or at times several thousand image and sound documents are available, and thus users can typically only see a fraction of the available data. To manage this complex and constantly changing database of images and sounds and to allow intuitive as well as creative data browsing, these systems were designed to deal with randomness and order, allowing partly directed and partly undirected searches. These systems create an interaction mode that is full of surprise, variety, dependency, irreducibility, connectivity, symmetry-breaking, low probability, information flow and information gain; this mode can most certainly be placed at the midpoint between order and disorder. These features are often associated with complex systems (see Section 2.4). While it could be argued that these systems simply use the Internet, which in itself is sometimes called a Complex System or even a CAS, it is apparent that through interaction certain data are triggered which in turn call and trigger other related data. All image and sound data streams in the above systems are unique and unrepeatable as the algorithms to retrieve these data work by randomizing the user's input and the search processes. This leads to constantly new search results, and the complex and constantly changing nature of the Internet itself adds additional noise and unpredictability. While we would not claim that the above systems are fully complex systems or CAS, we do propose that the randomizing of user input through multi-modal interaction and the randomizing of the search process can each serve as a means to create complexity features for systems on the Internet.

10.2 – Summary and Analysis of the Results

With "Life Spacies II" I have created an interactive artwork that satisfies key properties of complex systems, such as self-organization, self-reproduction, metabolisation, adaptation and evolution, reaction to its neighbours and to external control, exploration of its options, and the expansion of diversity. We can thus assume that this system qualifies as a Complex Adaptive System and that our goal of adapting CAS to interactive art has been achieved. Since the field of Complex Systems Sciences is not fully established and a comprehensive list of definitions or properties is not yet available, I also aimed to explore new options that create complexity outside the definitions of CAS. Exploring new territory, I thus introduced several experimental systems that aim to create or interact with complexity on the Internet. Here I tried to model properties of complex systems that are not covered by CAS. In these systems, the Internet was used as a platform for experimentation, and an interaction modus for interacting with complex and dynamically changing data on the Internet was created. While these systems cannot be called CAS, they might still be considered valuable as examples of how to model other forms of complexity and how to interact with an increasingly complex and dynamic database of on-line information.

10.3 - Future Research Directions and Possible Applications

My future research will concentrate on combining my efforts in creating real-time CAS for interactive art, implementing a learning function as a next step to enhance complexity, and exploring the application of these combined functions for an interactive system on the Internet. I also aim to design interfaces and interface protocols that allow for even more multimodal feedback and exchange between the users' input and the components of the system. Another area of research I aim to explore is porting principles of CAS to wireless networked devices and creating systems that can automatically adapt and organize themselves within complex databases of images, texts and sounds. As the amount of data on the Internet increases by the day, we will need more intelligent and adaptable search algorithms that can not only retrieve but also store user-dependent data, by for example anticipating a user's interests and modeling self-emerging weights of importance.

As practical art, I wish to create increasingly inter-connected interactive systems where increasingly complex and novel image and sound data can emerge through multi-modal interfaces that can themselves adapt, evolve, self-organize, reproduce and learn. These might be, for example, in the form of evolvable hardware systems as described by Thompson (1998).

11 – Acknowledgements

I would like to thank Prof. Roy Ascott, director of the CAiiA-Star research group at the University of Wales College of Art in Newport, Wales, UK, for making me part of his unique and highly creative international research community. The exchange with artists, theoreticians and researchers in this community has been a great inspiration for me. The regular research meetings and collaborative lectures we have given at conferences, symposia and universities around the word have provided a highly creative and critical study atmosphere that has helped me to express my artistic and scientific research endeavours more clearly. Dr. Michael Punt from the University of Wales College of Art in Newport, UK, has been a valuable discussion partner, and his critiques and advice have been essential for developing this thesis and for learning how to sharpen an argument. A very special thank you should also be extended to the administration of the University of Wales College of Art in Newport, Wales, UK, for supporting my thesis and for provding a research grant.

Prof. John Casti, my external supervisor, has not only been a long-time friend but also a great resource of information. Through on-line and face-to-face discussions with him I have been able to find relevant literature on complexity, complex systems, and origin of life theories. His own books on these topics are written in a superb style, and he is a master of making difficult scientific topics exciting and understandable for laymen readers. In my humble opinion, he is able to perfectly bridge the gap between performing cutting-edge scientific research and clearly explaining its relevance to the humanities, art, and society. I am very grateful that he took the time to advise me on these topics and to embark on the relatively unknown and experimental area of trying to combine complex system sciences with art.

At my home university, the IAMAS Institute of Advanced Media Arts and Sciences in Gifu, Japan, I would especially like to thank our president, Prof. Itsuo Sakane, who not only is a great friend and supporter but also gave me the opportunity to work in a highly creative and experimental media university, one of the most distinguished of its kind. I would also like to thank Professor Atsuhiro Sekiguchi, our principle teacher in Interactive Art, and all the professors and students of the IAMAS Institute of Advanced Media Arts and Sciences for allowing me extra time off to pursue my research. A very special thank you also goes to Mr. Ron Read of Kurdyla and Associates Co., Ltd. in Osaka, Japan, for final proofreading and correction of the manuscript.

The warmest thank you goes to my parents, especially to my mother Juliana, who was never tired of travelling to Japan or to many other places around the world to take care of my little son, Leo, while I worked on my thesis. Without her, this thesis would not have been possible. I also deeply thank my father Gunther (who after retirement at the age of 63 once again became a university student) for passing on to me his "gene for life-long study".

And a very, very special thanks goes to my partner Laurent Mignonneau, who has always supported me in the most selfless manner; through his artistic and technological skills and unlimited support I have been able to realise the artworks described in this thesis.

12 - Bibliography

Aaderaa, S.O. and Bröger, E. 1981. The equivalence of Horn and network complexity of Boolean functions. *Acta Informatica*, 15, pp. 303-307.

Ackley, D., and M. Littman 1992, Interactions between Learning and Evolution In Artificial Life II, edited by C. Langton, C. Taylor, J. Farmer, and S. Rasmussen. Santa Fe Institute Studies in the Sciences of Complexity Proc. Vol. X. Addison-Wesley, Redwood City, CA.

Alkon, D. L. 1989. Memory storage and neural systems, *Scientific American*, pp. 26-34.

Anderson, P.W. 1972. More is Different. *Science*, 177, pp. 393-396.

Annunziato, M. and Pierucci, P.. 2000. Relazioni Emergenti. In Artificial LifeVII Workshop Proceedings, Portland, Oregon pp. 162-165.

Arthur, B. 1993. Why do things become more complex? *Scientific American*, Vol. 268, pp. 92.

Ashby, W. Ross. 1962 Principles of the self-organizing system, in: Heinz Von Foerster & George W. Zopf, eds: Principles of Self-Organization. Pergamon Press, Oxford, pp. 255-278.

Ashby, W. Ross. 1969, Self-regulation and requisite variety. In Systems Thinking, edited by F.E. Emery, Penguin Books, Baltimore, pp. 105-124.

Ashby, W. Ross. 1972. Systems and their informational measures. In Trends in General Systems Theory, edited by G. J. Klir, John Wiley, New York, pp. 78-97.

Baas, N. A. 1994 Emergence, Hierarchies, and Hyperstructures, In C. G. Langton, ed., Alife III, Santa Fe Studies in the Sciences of Complexity, Proc. Volume XVII, Addison-Wesley, Redwood City, pp. 515-537.

Badii, R. and Politi, A. 1997. Complexity: hierarchical structures and scaling in physics. Cambridge University Press, Cambridge.

Bak, P. and Chen, K. 1991. Self-Organized Criticality. *Scientific American*. January 1991, pp. 46-54.

Bak, P., Tang, C. and Wiesenfeld, 1987. K. Self-organized criticality: An explanation of 1/f noise, *Phys. Rev. Lett.*, 59, pp. 381-384.

Baltschefsky, H. 1997. Major "Anastrophes" in the Origin and Early Evolution of Biological Energy Conversion. *J. Theoretical Biology*, 187, pp. 495-501.

Banzhaf, W. 1994. Self-Replicating Sequences of Binary Numbers: The Build-up of Complexity. *Complex Systems*, 8, pp. 215-225.

Bar-Yam, Y (ed.), 2000. Unifying Themes in Complex Systems: Proceedings of the First Necsi International Conference on Complex Systems, New York: Perseus Books.

Bateson, W. 1894. Materials for the Study of Variation. Cambridge: University Press.

Bedau, M. 1995. Three Illustrations of Artificial Life's Working Hypothesis. In Wolfgang Banzhaf and Frank Eeckman, eds., Evolution and Biocomputation, Berlin: Springer-Verlag, pp. 53-68.

Bedau, M. 1996a. The Nature of Life. In Boden, M., ed., The Philosophy of Artificial Life, Oxford: Oxford University Press, pp. 332-357.

Bedau, M., 1998. Philosophical Content and Method of Artificial Life, In T. W. Bynam and J. H. Moor, eds.,The Digital Phoenix: How Computers are Changing Philosophy, Basil Blackwell, pp. 135-152.

Bennett, CH. 1985. Dissipation, Information, Computational Complexity and the Definition of Organization. In PINES, D (Ed). Emerging Syntheses in Science. Redwood city, CA: Addison-Wesley, pp. 215-233.

Bennett, CH. 1988 Logical depth and physical complexity, In Rolf Herken, ed.: The Universal Turing Machine. Oxford University Press, Oxford, pp. 227-257.

Bennett, CH. 1990. How to Define Complexity in Physics, and Why. In Zurek, WH (Ed). Complexity, Entropy and the Physics of Information. Redwood City California: Addison-Wesley, pp. 137-148.

Biles, J. A. 1994. GenJam: A genetic algorithm for generating jazz solos. In ICMC Proceedings. The Computer Music Association.

Bilotta, E., Pantano, P. and Talarico, V. 2000. Synthetic Harmonies: an approach to musical semiosis by means of cellular automata, in Artificial Life VII Proceedings of the Seventh International Conference, edited by M.A. Bedau, J.S. McCaskill, N. H. Packard, and St. Rasmussen, (Boston: MIT Press, 2000), pp. 537-546

Biota, 2001. Home page: http://www.biota.org/

Blomberg, C., Liljenström, H. Lindahl, B.I.B. and Århem, P. eds. 1994. Mind and matter: essays from biology, physics and philosophy. *J. Theoretical Biology*, 171, pp. 1-122.

Blum, M. 1967. A Machine Independent Theory of the Complexity of Recursive Functions. *Journal of the Association of Computing Machinery*, Vol. 14, pp. 322.

Boden, M. 1996.The Philosophy of Artificial Life (Oxford Readings in Philosophy.) New York: Oxford University Press.

Böhler, C., Nielsen, P.E. and Orgel, L.E. 1995. *Nature* Vol.376. pp. 578-581.

Bynam, T.W. and Moor, J. H. 1998. The Digital Phoenix: How Computers are Changing Philosophy, Malden: Blackwell Publishers, pp. 135-152.

Cariani, P.. 1992 "pEmergence and Artificial Life"p, In Christopher G. Langton, Charles Taylor, J. Doyne Farmer and Steen Rasmussen, eds.: Artificial Life II. Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. X. Addison-Wesley, Redwood City, Calif., pp. 775-797.

Cairns-Smith, A.G. 1982. Genetic Takeover. Cambridge: Cambridge University Press.

Cairns-Smith, A.G. 1986. Clay Minerals and the Origin of Life. Cambridge: Cambridge University Press.

Casti, J.L. 1979.Connectivity, Complexity and Catastrophe. NY: John Wiley.

Casti, J.L. and Karquist, A. 1985. (Eds.). Complexity, Language and Life: Mathematical Models. Berlin: Springer-Verlag.

Casti, J.L. 1986. On System Complexity: Identification, Measurement and Management. In J. Casti & A. Karlquist (eds.) Complexity Language and Life: Mathematical Approaches. Berlin: Springer, pp. 146-173.

Casti, J.L. 1992. The simply complex: trendy buzzword or emerging new science? Bulletin of the Santa Fe Institute, Vol. 7, pp. 10-13.

Casti, J.L. 1994. Complexification. London: Abacus.

Casti, J.L.1996. What if. *New Scientist*, Vol. 151/2038, pp. 36-40.

Casti, J.L. 2000. Paradigms Regained. New York: William Morrow & Co.

Ceccato, C. 2000, On the Translation of Design Data into Design Form in Evolutionary Design, in: Generative Art 2000 Conference Proceedings.

Chaitin, G. J. 1966. On the Length of Programs for Computing Finite Binary Sequences. *Journal of the Association of Computing Machinery*, 13, pp. 547-569.

Chaitin, G.J. 1992 Information Theoretic Incompleteness. World Scientific, Singapore.

Chomsky, N. 1972. Language and Mind, New York: Hbj College & School Div.

Churchland, P. S. 1986. Neurophilosophy: Toward a Unified Science of the Mind/Brain, Cambridge: Bradford Books/MIT Press.

Churchland, P. M. 1989. A Neurocomputational Perspective: The Nature of Mind and the Structure of Science, Cambridge: Bradford Books/MIT Press.

Chyba, C. 1997. A Left-Handed Solar System? *Nature*, Vol.389, pp. 234-235.

Collins, R. J., and D. R. Jefferson 1992, AntFarm: Towards Simulated Evolution, In Artificial Life II, edited by C. Langton, C. Taylor, J. Farmer, and S. Rasmussen. Santa Fe Institute Studies in the Sciences of Complexity Proc. Vol. X. Addison-Wesley, Redwood City, CA.

Conrad, M., and Strizich, M. 1985, EVOLVE II: A Computer Model of an Evolving Ecosystem, In *Biosystems* 17, pp. 245-258.

Cornacchio, J.V. 1977. Maximum Entropy Complexity Measures. *International Journal of General Systems*, Vol. 3, pp. 267-271.

Crick, F.H.C. 1968. The origin of the genetic code. *J. Molecular Biology*, Vol. 38, pp. 367-379.

Crick, F.H.C., Brenner, S., Klug, A. and Pieczenik, G. 1976. A speculation of the origin of protein synthesis. *Origins Life*, Vol. 7, pp. 389-397.

Crutchfield, J.P., Farmer, J.D., Packard, N.H., and Shaw, R. S. 1986. Chaos. *Scientific American*, 255 (December), pp. 46-57.

Crutchfield, J.P. 1993. Observing Complexity and The Complexity of Observation. In Atmanspacher, H (Ed). Inside versus Outside. Berlin: Springer, pp. 235-272.

Crutchfield, J.P. 1994. The Calculi of Emergence: Computation, Dynamics and Induction. *Physica* D, 75, pp. 11-54.

Cruz-Neira, C., Sandin, D. and DeFanti, T. 1993. Virtual Reality: The Design and Implementation of the CAVE, In proceedings of SIGGRAPH 93 Computer Graphics Conference, ACM SIGGRAPH, pp. 135-142.

Dahlstedt, P. and Nordahl, M.G. 1999. Living Melodies: Coevolution of Sonic Communication. In: First Iteration Conference on Generative Processes in the Electronic Arts, Melbourne, Australia.

Damer, B., Marcelo, K. and Revi, F. 1998. Nerve Garden: a Public Terrarium in Cyberspace, *LEONARDO Journal*, Vol. 31, No. 5, p. 385.

Daniels, M. 1999. An open framework for agent-based modeling. Lecture Notes at University of Chicago, available at: http://www.swarm.org

Darwin, C. 1959 [1859].The Origin of Species. Reprinted. London: Penguin.

Dawkins, R. 1986. The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe without Design. London: Longman.

Dawkins, R. 1986. The Blind Watchmaker, W.W. Norton, The Blind Watchmaker. Appendix: BIOMORPHS - Evolution Simulation Software.

Dawkins, R. 1989. The evolution of evolvability. In C.G.Langton (Ed.). *Artificial Life*. Reading, MA: Addison-Wesley.

De Duve, C. 1991. The Beginnings of Life on Earth. *American Scientist*, Vol. 83, pp. 428-437.

Dennett, D. 1994. Artificial Life as Philosophy. Artificial Life 1(3), pp. 291-292.

Dennett, D.. 1995. Darwin's Dangerous Idea: Evolution and the Meanings of Life, Simon and Schuster: New York.

Dooley, K., Johnson, T., and Bush, D. 1995. TQM , Chaos, and Complexity, *Human Systems Management*, Vol. 14, pp. 1-16.

Dorin, A. 1994. A Model of Protozoan Movement for Artificial Life, in Insight Through Computer Graphics, Proceedings of CGI94: Computer Graphics International, World Scientific Press, pp. 28-38.

Dorin, A. 1999. Classification of Physical Processes for Virtual-Kinetic Art, in Proceedings of First Iteration, Dorin & McCormack (eds), CEMA, Melbourne.

Dorin, A. and McCormack, J. (eds.) 1999. First Iteration: Proceedings of the First International Conference on Generative Systems in the Electronic Arts, CEMA, Melbourne.

Driesch H. 1914. The History and Theory of Vitalism. London: Macmillan.

Edmonds, B. 1999. Syntactic Measures of Complexity. Doctoral Thesis, University of Manchester, Manchester, UK.

Eigen, M. and Winler-Oswatitch, R. 1992. Steps Towards Life: A Perspective on Evolution. New York: Oxford University Press.

Emmeche, C. 1992. Life as an Abstract Phenomenon: Is Artificial Life Possible? in: Francisco J. Varela and Paul Bourgine (eds.):Toward a Practice of Autonomous Systems. Proceedings of the First European Conference on Artificial Life. The MIT Press, Cambridge, Mass., pp. 466-474.

Emmeche, C,, Koppe, S. and Stjernfelt, F. 1997. Explaining Emergence: towards an ontology of levels, In *Journal for General Philosophy of Science*, 28, pp. 83-119.

Evolva, 2000. Computer Artworks Corportation, http://www.artworks.co.uk/evolva.htm

Fagerlund, M. 2001. Home Page: http://www.cambrianart.com/

Fagin, R. 1973. Contributions to the Model Theory of Finite Structures, Ph.D.Thesis, University of California Berkeley.

Ferdinand, A.E. 1974. Complex systems are those that work even with maximum entropy. International Journal of General Systems, Vol. 1, pp. 19-33.

Fogel, L. J. , Owens, A. J. and Walsh, M. J. 1966. Artifcial Intelligence Through Simulated Evolution. John Wiley & Sons, Inc.

Fontana, W. 1992. Algorithmic Chemistry. In: C.G. Langton, J.D. Farmer, S. Rasmussen and C. Taylor (eds.) Artificial Life II: Santa Fe Institute Studies in the Sciences of Complexity. Vol. 10. Reading, MA: Addison-Wesely, pp. 159-209.

Forrest, S. and T. Jones. 1994. Modeling complex adaptive systems with Echo. In R.J. Stonier and X.H. Yu, eds. Complex Systems: Mechanisms of Adaptation. IOS Press, Amsterdam., pp. 3-21.

Freude, A. C.H. and Espenschied, D. 1999. Assoziations-Blaster!, http://www.assoziations-blaster.de/

Gaffney, N. 1998. Mutagen. http://rorschach.test.at/mutagen/

Gánti, T. 1979. A Theory of Biochemical Supersystems and its Application to Problems of Natural and Artificial Biogenesis. Baltimore, MD: University Park Press.

Gardner, M. 1970. Mathematical Games: The Fantastic Combinations of John Conway's New Solitaire Game 'Life', *Scientific American*, 223(4), pp. 120-123.

Gardner, M. 1984. The Sixth Book of Mathematical Games from Scientific American. Chicago, IL: University of Chicago Press, p. 91.

Gell-Mann, M 1994. The Quark and the Jaguar - adventures in the simple and the complex. London: Little, Brown and Co.

Gell-Mann, M. 1995. What is Complexity? *Complexity*, Vol. 1, pp. 16-19.

Gell-Mann, M and Lloyd, S. 1996. Information Measures, Effective Complexity and Total Information. *Complexity*, Vol. 2, pp. 44-52.

Generative Art. 1998-2001, International Conference on Generative Art, Politecnico di Milano University, Milan, Italy, http://www.generativeart.com/

George, B. 1977. Tests for System Complexity. *International Journal of General Systems*, Vol. 3, pp. 253-257.

Gilbert, W. 1986. The RNA World. *Nature*, Vol. 319, p. 618.

Girard, J.Y. 1987. Proof Theory and Logical Complexity. Naples: Bibliopolis.

Golay, M.W.,Seong, P.H., Manno, V.P. 1989. A Measure of the Difficulty of System Diagnosis and its Relationship to Complexity. *International Journal of General Systems*, Vol. 16, pp. 1-23.

Goldberg, D.E. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Reading. Mass.: Addison-Wesley.

Goodman, N. 1966. The Structure of Appearence. Indianapolis: Bobbs-Merrill.

Goodwin, B. 1994. How the Leopard Changed its Spots. London: Orion Books (Phoenix), p. 169.

Gould, Stephen Jay. 1989. Wonderful Life: The Burges Shale and the Nature of History, New York: Norton.

Grand, S., Cliff, D., Malhotra, A. 1996. Creatures: artificial life autonomous software agents for home entertainment. In: Millennium Technical Report 9601, University of Sussex Technical Report CSRP434.

Grand, S., Cliff, D., Malhotra, A. 1997. Creatures: Artificial Life Autonomous Software Agents for Home Entertainment, Proceedings of the First International Conference on Autonomous Agents, ACM Press, New York, pp. 22-29.

Grassberger, P. 1986. Towards a Quantitative Theory of Self-Generated Complexity. *International Journal of Theoretical Physics*, 25, pp. 907-938.

Grassberger, P. 1989. Problems in Quantifying Self-organized complexity. *Helvetica Physica Acta,* Vol. 62, pp. 498-511.

Green, D.G. 1993. Emergent Behaviour in Biological Systems. In Green, DG; Bossomaier, TJ (Eds). Complex Systems - From Biology to Computation. Amsterdam: OS Press, pp. 24-35.

Griffeath, D., Fisch R. and Gravner J. 1991. Cyclic Cellular Automata in Two Dimensions, In: Spatial Stochastic Processes: A festschrift in honor of the seventieth birthday of T.E. Harris, K.S. Alexander and J.C. Wadkins eds. Birkhäuser, Boston, pp. 171-185.

Haken, H. 1988. Information and Self-Organisation: a macroscopic approach to complex systems. Berlin: Springer-Verlag, Springer Series in Synergetics.

Haldane, J.B.S. 1932. The Forces of Evolution. London: Longmans Green.

Han, J. and Kamber M. 2000. Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers.

Harary, F. 1994. Graph Theory. Reading, MA: Addison-Wesley.

Hayek, F.A. 1964. The Theory of Complex Phenomena. In BUNGE, M (Ed). The Critical Approach to Science and Philosophy. London: Collier McMillan, pp. 332-349.

Hebb, D. O. 1949, The Organization of Behavior, John Wiley and Sons, Inc.

Heudin, J. C., 1998. L'évolution au bord du chaos - Préface de Joël de Rosnay, Editions Hermès Science, Paris.

Heylighen, F. 1996. The Growth of Structural and Functional Complexity during Evolution, in F. Heylighen & D. Aerts (eds.) The Evolution of Complexity, Kluwer Academic Publishers.

Hillis, D. 1992, Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure, In Artificial Life II, edited by C. Langton, C. Taylor, J. Farmer, and S. Rasmussen. Santa Fe Institute Studies in the Sciences of Complexity Proc. Vol. X. Addison-Wesley, Redwood City, CA, pp. 313-324.

Hinegardner, R. and Engelberg, H. 1983. Biological Complexity, *J. Theoret. Biol.*, 104, pp. 7-20.

Holland, J. H. 1975. Adaptation in Natural and Artifcial Systems. AnnArbor: University of Michigan Press.

Holland, J.H. 1994. Echoing emergence: Objectives, rough definitions, and speculations for Echo--class models. in G.A. Cowan, D. Pines and D. Meltzer, eds. Complexity: Metaphors, Models and Reality. Addison Wesley, Reading, MA.

Hoyle, F. and Wickramasinghe, N.C. 1979. Diseases from Space. New York: Harper and Row.

Hraber, P., Jones, T. and Forrest, S. 1997. The Ecology of Echo, in: Artificial Life 3: MIT Press, pp. 165–190.

Hurry, M., Prophet, J., Selley, G. 2000. TechnoSphere. In Artificial LifeVII Workshop Proceedings, Portland, Oregon pp. 154-157.

Innocent, T. 2001. The Language of Iconica. *LEONARDO Journal*, (Cambridge/MA: MIT Press), Vol. 34, No. 3., pp. 258-264.

Inoue, J. 1992. Asymmetric Photochemical Reactions in Solution. *Chemical Reviews*, Vol.92, pp.741-770.

International Conference on Generative Art and Design, 2000. Home Page: http://www.generativeart.com

Internet Explorer, 2001. http://www.microsoft.com/windows/IE/

Ishihama, Y. (Stone), 2000. Home Page: http://www.asahi-net.or.jp/~hq8y-ishm/index.html

Jantsch, Erich. 1980. The Self-Organizing Universe. Pergamon, Oxford and New York.

Jefferson D., Collins R., Cooper C., Dyer M., Flowers M, Korf K., Taylor C. and Wang, A. 1992. Evolution as a Theme in Artificial Life: The Genesys/Tracker System In Artificial Life II, edited by C. Langton, C. Taylor, J. Farmer, and S. Rasmussen. Santa Fe Institute Studies in the Sciences of Complexity Proc. Vol. X. Addison-Wesley, Redwood City, CA. pp. 417-434.

Johnston, R. B. 1996. From Efficiency to Flexibility: Entropic Measures of Market Complexity and Production Flexibility. In Complex 96: From Local Interactions toGlobal Phenomena, Charles Sturt University, Albury, NSW, IOS Press, Amsterdam. pp. 30 - 140.

Kampis, G. 1991. Self-Modifying Systems in Biology and Cognitive Science: A New Framework for Dynamics, Information and Complexity. Oxford: Pergamon Press.

Karjalainen, S, Kyrölä, A. et al. 2000. Habbo Hotel (original title: Hotelli Kultakala), http://www.habbohotel.co.uk

Kauffman, St. 1993 The Origins of Order. Self-organization and Selection in Evolution. Oxford University Press, Oxford.

Kauffman, St. 1995. At Home in the Universe. The Search for Laws of Complexity. New York: Oxford University Press, pp. 23 - 112.

Kellert, S. H. 1993. In the Wake of Chaos: Unpredictable Order in Dynamical Systems, Chicago: The University of Chicago Press.

Kelly, G.A. 1955. The Psychology of Personal Constructs, New York: Norton.

Kemeny, J.G. 1953. Two Measures of Complexity. *The Journal of Philosophy*, 52, pp. 722-733.

Kerne, A. 1997. CollageMachine: Temporality and Indeterminacy in Media Browsing via Interface Ecology, In Human Factors in Computing Systems: Extended Abstracts of CHI 97, Atlanta, Georgia.

Kerne, A. 2001. CollageMachine: A Model of "Interface Ecology," Ph.D. Thesis, Department of Computer Science, New York University.

Khalil, E.l. 1995. Organizations, Naturalism and Complexity. *Review of Social Economy*, 53, pp. 393-419.

Kitcher, P. 1984. 1953 All That: A Tale of Two Sciences. *Philosophical Review* 93, pp. 335-373.

Kleinberg., J. 2000. Navigation in a Small World. *Nature*, 406, 845.

Kleiweg, P. 1998. Genetic Art, http://odur.let.rug.nl/~kleiweg/genart/genart.html

Kolmogorov, A.N. 1965. Three Approaches to the quantitative definition of Information, *Problems of Information Transmission*, 1, pp. 1-17.

Koza, J R 1992. Genetic Programming: On the programming of computers by means of natural selection. Cambridge, Mass.: MIT Press.

Kusahara, M. 2000. The Art of Creating Subjective Reality: An Analysis of Japanese Digital Pets. In: Artificial LifeVII Workshop Proceedings, Portland, Oregon, pp. 141-144.

LaFarge, A., 2000. WinSide Out. Essay for the Shift_Control exhibition at the Bealls Art Center, University of California at Irvine. available at: http://beallcenter.uci.edu/shift/homens.html

Landauer, R. 1988. A simple measure of complexity, *Nature* 336, pp. 306-307.

Langton, C. G., eds. 1989 Artificial Life. Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol.6. Redwood City, Calif.: Addison-Wesley.

Langton, C. 1989. Artificial Life, in C. Langton, ed., Artificial Life, Redwood City: Addison-Wesley, pp. 1-47.

Langton, C., Taylor, C., J. Farmer, D. and Rasmussen, St. eds.1992 Artificial Life II. Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. X. Redwood City, Calif.: Addison-Wesley.

Langton, C. 1992. Life at the Edge of Chaos. In C. Langton, C. Taylor , D. Farmer, and S. Rasmussen, eds., Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity, Vol. X, Redwood City: Addison-Wesley, pp. 41-91.

Langton, C., Minar, N. and Burkhart, R. 1995. The Swarm Simulation System: A Tool for Studying Complex Systems. Santa Fe Institute Working Papers. and at: http://www.santafe.edu/projects/swarm.

Lewes, G. H. 1874-75 Problems of Life and Mind, first series, 2 vols. Truebner and Co., London. [third series, 1879].

Lewin, R. 1992. Complexity: Life at the Edge of Chaos. London: Macmillan Publishing, pp. 178.

Levine, R.D and Tribus, M (Eds.). 1979. The Maximum Entropy Principle. Cambridge, Massachusetts: MIT Press.

Life Drop, 2000. available at: http://www.virtual-worlds.net/lifedrop/

Lopez, L.R. and Caufield, L.J. 1991. A Principle of Minimum Complexity in Evolution. Lecture Notes in Computer Science, Vol. 496, pp. 405-409.

Marcus, M.J. 1977. The Theory of Connecting Networks and their Complexity: A Review. *Proceedings of the IEEE*, Vol. 65, pp. 1263-1271.

Margaleff, R. 1984. Ecosystems: Diversity and Connectivity as measurable components of their complication. In AIDA, et al (Ed). The Science and Praxis of Complexity. Tokyo: United Nations University, pp. 228-244.

Martinez, J.L. 1995. On Complexity and Simplicity as Different Evolutionary Strategies. *Evolutionary Theory*, Vol. 10, pp. 317-320

Maturana, H. R., and Varela, F. J. 1980. Autopoiesis and Cognition: The Realisation of the Living. London: Reidel.

Maynard Smith, J. and Szathmár, E. 1995. The Major Transitions in Evolution. Oxford: W.E. Freeman.

McCormack, J. 1994a, Turbulence, In Visual Proceedings of SIGGRAPH 94, ACM, New York.

McCormack, J. 1996. Grammar Based Music Composition, in Complex Systems 96: From Local Interactions to Global Phenomena, R. Stocker et. al., eds, ISO Press, Amsterdam.

McCulloch, W. and Pitts, W. 1943. A Logical Calculas of the Ideas Immanent in Nervous Activity, Bulletin of Mathematical Biophysics 5, pp. 115-133

McShea, D. W. 1996. Metazoan complexity and evolution: Is there a trend? *Evolution* 50, pp. 477-492.

Meyer, A.R. and Ritchie, D.M. 1967. The Complexity of Loop Programs. In Proceedings of the 22nd National Conference. Washington DC: Thompson Book Company, pp. 465-469.

Mignonneau, L. and Sommerer, C., 2000. Designing Interfaces for Interactive Artworks, in KES 2000 Knowledge Based Engineering Systems Conference Proceedings (University of Brighton, UK, 2000), pp. 80-84.

Mignonneau, L., Sommerer, C. and Lopez-Gulliver, R. 2001a. Riding the Net: a Novel, Intuitive and Entertaining Tool to Browse the Internet, In Siggraph 2001 Conference Proceedings, ACM Siggraph, p. 133.

Mignonneau, L., Sommerer, C., Lopez-Gulliver, R. and Jones, S. 2001b. Riding the Net: a Novel, Intuitive and Entertaining Tool to Browse the Internet, in SCI 2001- 5th World Multiconference on Systemics, Cybernetics and Informatics Conference Proceedings, (Orland, Florida: International Institute of Informatics and Systemics, 2001), pp. 57-63.

Miller, S.L. 1953. Production of amino acids under possible primitive earth conditions. *Science*. 117: 528.

Miller, G. F., and P. M. Todd 1991, Exploring Adaptive Agency I: Theory and Methods for Simulating the Evolution of Learning, In Connectionist Models, Proceedings of the 1990 Summer School, edited by D. S. Touretzky, J. L. Elman, T. J. Sejnowski, G. E. Hinton, Morgan Kaufmann, San Mateo, CA, 1991.

Miranda, E. B, 2000. Modelling the Evolution of Complex Sounds with Cellular Automata, In: Creative Evolutionary Systems, Peter J Bentley and David W Corne (eds.), Morgan Kaufmann.

Mitchell, M., Crutchfield, J. P., Hraber, P. T., 1994 Evolving Cellular Automata to Perform Computations: Mechanisms and Impediments, *Physica* D, 75, pp. 361-391.

Mitchell, M. and Forrest, S. 1997. Fitness Landscapes: Royal Road Functions. In Back, Fogel, and Michalewicx (Eds.) Handbook of Evolutionary Computation. Institute of Physics Publishing, Phiuladelphia and Bristol UK, B2.7. pp. 1-25.

Moore, J.H. 1994. GAMusic: Genetic algorithm to evolve musical melodies. Windows 3.1 software available at: http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/genetic/ga/systems/ gamusic/0.html.

Moreno, A. 2000. Artificial Life as a bridge between Science and Philosophy. In: Artificial Life VII: Proceedings of the Seventh International Conference, edited by Mark A. Bedau, John S. McCaskill, Norman H. Packard, and Steen Rasmussen (MIT Press, 2000), pp. 507-512.

Morgan, C. L. 1923. Emergent Evolution. Williams and Norgate, London.

Morowitz, H. 1992. Beginnings of Cellular Life. New Haven, CT: Yale University Press.

Moshowitz, A. 1968. Entropy and the Complexity of Graphs. *Bulletin of Mathematical Biophysics*, Vol. 30, 175-204, 225-40.

Mount, J. 1998. Genetic Art III, http://www.cs.cmu.edu/Web/People/jmount/g3.html

Musgrave, K. 1998. Genetic Programming, Genetic Art: Dr. Mutatis' at http://www.seas.gwu.edu/faculty/musgrave/mutatis.html

Nadenau, M.J. and Reichel, J. 2000. How to measure arithmetic complexity of compression algorithms: A simple solution. In Proceedings of ICME2000, New York, New York, Jul. 30-Aug. 2000, pp. 1743-1746

Nelson, G. L. 1993. Sonomorphs: An application of genetic algorithms to the growth and development of musical organisms. In Proceedings of the Fourth Biennial Art and Technology Symposium, Connecticut College, pp. 155-169.

Nelson, R.J. 1976. Structure of Complex Systems. *Philosophy of Science Association*, 2, pp. 523-542.

Netscape, 2001. www.netcape.com

Nicolis, G. and Prigogine, I. 1989. Exploring Complexity. San Francisco: W. H. Freeman & Co.

Nielsen P. E., Egholm, M., Berg, R.H. and Buchardt, O. 1991. *Science*. Vol. 254, pp. 1497-1500.

Oparin, A. I. 1924 [translated to English 1938, new edition 1957]. The Origin of Life on Earth. New York: Academic Press.

Orgel, L.E. 1987. The Origin of Self-Replicating Molecules. In: Self-Organizing Stystems: The Emergence of Order. (F.E. Yates, ed.), New York: Plenum Press, pp. 65-74.

Owen, R. 1980 [1861]. Paleontology or a Systematic Summary of Extinct Animals and Their Geological Relation. London: Ayer Co. Publisher.

Packard, N. 1989, Intrinsic Adaptation in a Simple Model for Evolution, in Artificial Life, edited by C. Langton, Santa Fe Institute Studies in the Sciences of Complexity Proc. Vol. VI. Addison-Wesley, Redwood City, CA.

Pagels, H. 1988 The Dreams of Reason. Simon & Shuster (Bantam ed., N.Y. 1989).

Papadimitriou, Ch.H. 1994. Computational Complexity. Addison Wesley.

Pattee, H. 1977. Dynamic and Linguistic Modes of Complex Systems. *International Journal of General Systems*, 3, pp. 259-266.

Pincus, S. 1995. Approximate entropy (ApEn) as a complexity measure. *Chaos*, Vol. 5, pp. 110-117.

Poli, R. and Cagnoni, S. 1997. Evolution of Pseudo-colouring Algorithms for Image Enhancement with Interactive Genetic Programming, Birmingham EEBIC group Technical Report CSRP-97-5.

Pontecorvo, M. 2000. Designing the Undesigned: Emergence as a tool for design, In Generative Art 2000 Conference Proceedings.

Prigogine, Ilya, and Stengers, Isabelle. 1997. The End of Certainty, Time, Chaos and the New Laws of Nature, New York: The Free Press.

Prusinkiewicz, P., and Lindenmayer, A., eds. 1990. The Algorithmic Beauty of Plants. New York: Springer Verlag.

Ray, T. S. 1991. An approach to the synthesis of life. In Langton, C., C. Taylor, J. D. Farmer, & S. Rasmussen [eds], Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity, vol. XI, Redwood City, CA: Addison-Wesley, pp. 371-408.

Ray T.S. 1994. Evolution, Complexity, Entropy and Artificial Reality. *Physica* D, Vol. 75, pp. 239-263.

Rechenberg, I. 1998. Cybernetic solution path of an experimental problem. Royal Aircraft Establishment, Library Translation 1122, 1965. Reprinted in `Evolutionary Computation | The fossil record', D. B. Fogel, ed., IEEE Press. chap. 8, pp 297-309.

Reynolds, C. W. 1987. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*. Vol 21, No 4, pp. 25-34.

Reynolds, C. 2001. Home Page: Evolutionary Computation and its application to art and design, http://www.red3d.com/cwr/evolve.html

Riddell, A. M. 1999. Data Culture Generation, Proceedings of the First Iteration - A conference on Generative Processes in Electronic Arts. Monash University December 1999.

Ritz, G., 2002. On-line Discussion on Complexity: http://necsi.org/discuss/discuss.html

Rook S, 2000. Eons of Genetically Evolved Algorithmic Images - Steven Rook, in: Creative Evolutionary Systems, Peter J Bentley and David W Corne (eds.), Morgan Kaufmann.

Rooke, S. 1998. Genetic Art Process. http://www.concentric.net/~srooke/process.html

Rosen, R. 1977. Complexity and system descriptions. In W.E. Harnett (ed.) Systems: Approaches, Theories, Applications. Dordrecht, Holland: D. Reidel, pp. 169-175.

Rouse, W.B. and Rouse, S.H. 1979. Measures of Complexity of Fault Diagnosis Tasks. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 9, pp. 720-727. (Reprinted in B. Curtis, Ed., 1981, Human Factors in Software Development. Silver Spring, MD: IEEE Computer Society Press.)

Rowbottom, A. 1998. Organic, Genetic and Evolutionary Art. http://www.netlink.co.uk/users/snaffle/form/evolutio.html, 10 August 1998.

Rowland. D. 1998. Computer Graphic Control over Human Face and Head Appearance, Genetic Optimisation of Perceptual Characteristics, PhD thesis, University of Hull.

Salton, G. 1983. Introduction to Modern Information Retrieval, New York: McGraw Hill.

Saunders, PT and Ho, M.W. 1981. On the Increase in Complexity in Evolution. *Journal of Theoretical Biology*, Vol. 90, pp. 515-530.

Savage, J.E. 1987. The Complexity of Computing. Krieger Publishing Company.

Schwefel, H.-P. and Rudolph, G. 1995. Contemporary evolution strategies. In F. Morgan, A. Moreno, J. J. Merelo, and P. Chacon, editors, Advances in Arti¯cial Life: Proc. 3rd Eur. Conf. on Artifcial Life, volume 929 of LNAI, Springer-Verlag, pp. 893-907.

Shannon, C. E. 1948. A Mathematical Theory of Communication, *Bell System Technical Journal*, Vol. 27, pp. 379-423 and 623-656.

Shannon, C.E and Weaver, W. 1949. The Mathematical Theory of Communication. Urbana, Illinois: University of Illinois Press.

SimLife, 1992. Maxis Software. http://thesims.ea.com/

Sims, K. 1991. Artificial Evolution for Computer Graphics, in *Computer Graphics*, Vol.25, 4, pp. 319-328.

Sims, K. 1994a Evolving Virtual Creatures, In *Computer Graphics* (Siggraph '94) Annual Conference Proceedings, July 1994, pp.43-50.

Sims, K., 1994b. Evolving 3D morphology and behavior by competition, In Artifcial Life IV, R. Books and P. Maes (ed.), MIT Press, Cambridge, MA, pp. 40-48.

Smith, C.U.M. 1990. Elements of Molecular Neurobiology, Chichester: Wiley

Smith, C.U.M. 1995. The Complexity of Brains: A Biologist's View. *Complexity International*, Vol. 2.

Soddu, C.. 2000. From Forming to Transforming, In Generative Art 2000 Conference Proceedings.

Solomonoff, R.J. 1964. A Formal Theory of Inductive Inference. *Information and Control*, 7, 1-22,pp. 224-254.

Sommerer, C. and Mignonneau, L. 1993. Interactive Plant Growing, in Genetische Kunst - Kuenstliches Leben, Ars Electronica 93, PSV Verlag, 1993, pp. 408-414.

Sommerer, C. and Mignonneau, L. 1994. A-Volve: A real-time interactive environment, In ACM Siggraph Visual Proceedings, New York, pp. 172-173.

Sommerer, C. and Mignonneau, L. 1995. Trans Plant. In Imagination. T. Moriyama (ed), Tokyo Metropolitan Museum of Photography, Chapter 2.

Sommerer, C. and Mignonneau, L. 1996. MIC Exploration Space. In Siggraph'96 Visual Proceedings, ACM Siggraph, New York, p. 17.

Sommerer, C. and Mignonneau, L. 1997a. Interacting with Artificial Life: A-Volve, *Complexity Journal*, (New York: Wiley, 1997), Vol. 2, No. 6, pp. 13-21.

Sommerer, C. and Mignonneau, L. 1997b. A-Volve: an evolutionary artificial life environment. In: C. Langton and K. Shimohara (eds.) AlifeV: Proceedings of the Artificial Life V, (Boston: MIT Press), pp. 167-175.

Sommerer, C. and Mignonneau, L. 1997c. Life Spacies. In ICC Concept Book, NTT-ICC, Tokyo, pp. 96 P101.

Sommerer, C. and Mignonneau, L. 1998a. The application of artificial life to interactive computer installations, *Artificial Life and Robotics Journal*, Springer Verlag Tokyo, Vol.2, No.4, pp. 151-156.

Sommerer, C. and Mignonneau, L. 1998b. Art as a Living System, In Art @ Science, C. Sommerer & L. Mignonneau, (Eds.), Wien/New York: Springer Verlag, pp. 148-161.

Sommerer, C. and Mignonneau, L. 1999a. Art as a Living System, *LEONARDO Journal*, (Cambridge/MA: MIT Press), Vol. 32, No. 3., pp. 165-173.

Sommerer, C. and Mignonneau, L. 1999b. Life Spacies, In Siggraph'99 Conference Abstracts and Applications, (New York: ACM Siggraph), p. 170.

Sommerer, C. and Mignonneau, L. 1999c. Life Spacies: a genetic text-to-form editor on the internet, In AROB'99 International Symposium on Artificial Life and Robotics Conference Proceedings (Oita University, Japan) pp. 73-77.

Sommerer, C. and Mignonneau, L. 1999d. Text as Genetic Code: Creating Visual Language on the Internet, In CADE'99 Computers in Art and Design Conference Proceedings (University of Teeside UK), Appendices.

Sommerer, C. and Mignonneau, L. and Lopez-Gulliver, R. 1999e. LIFE SPACIES II: from text to form on the Internet using language as genetic code. In Proceedings ICAT'99 9th International Conference on Artificial Reality and Tele-Existence, (Tokyo: Virtual Reality Society), pp. 215-220.

Sommerer, C. and Mignonneau, L. 1999f. VERBARIUM and LIFE SPACIES: Creating a Visual Language by Transcoding Text into Form on the Internet, In IEEE Symposium on Visual Languages Conference (VL'99) Conference Proceedings (Tokyo), pp. 90-95.

Sommerer, C. and Mignonneau, L. 1999g. VERBARIUM - Interactive Website; Collection of the Cartier Foundation, Paris; http://www.fondation.cartier.fr/verbarium.html.

Sommerer, C. and Mignonneau, L. 2000a. Modeling Emergence of Complexity: the Application of Complex System and Origin of Life Theory to Interactive Art on the Internet, In Artificial Life VII, edited by M.A. Bedau, J.S. McCaskill, N. H. Packard, and St. Rasmussen, (Boston: MIT Press, 2000), pp. 547-554.

Sommerer, C. and Mignonneau, L. 2000b. Life Spacies II, In Ars Electronica 2000, (Vienna/New York: Springer Verlag), pp. 392-392.

Sommerer, C. and Mignonneau, L. 2000c. Modeling Complex Systems for Interactive Art, In Applied Complexity - From Neural Nets to Managed Landscapes, edited by S. Halloy and T. Williams (Institute for Crop & Food Research, Christchurch, New Zealand,), pp.25-38.

Sommerer, C. and Mignonneau, L. 2000d. Modeling Complex Systems for Interactive Art on the Internet, In MMM2000 MultiMediaModeling Conference Proceedings Nagano, Japan, (World Scientific), pp. 237-254.

Sommerer, C. 2001a. ALife in Art, Design, Edutainment, Game and Research, *LEONARDO Journal*, (Cambridge/MA: MIT Press), Issue 34:4, August 2001, pp. 297-298.

Sommerer, C., Mignonneau, L. and Lopez-Gulliver, R. 2001b. The Living Room. Presented at the Bo01 - Living in the Future exhibition. Malmoe, Sweden. May 2001.

Sommerer, C. and Mignonneau, L. 2001c. Creating and Communicating with Virtual Life on Mobile Phones," In: SCI 2001- 5th World Multiconference on Systemics, Cybernetics and Informatics Conference Proceedings, (Orland, Florida: International Institute of Informatics and Systemics), pp. 64-70.

Sommerer, C., Mignonneau, L. and Lopez-Gulliver, R. 2001d. IKI-IKI Phone: a Multi-user Alife Art Game for Mobile Phones, In IEEE International Conference on Multimedia and Expo (ICME2001) Conference Proceedings, (Tokyo: Waseda University, 2001), CDRom Proceedings.

Sommerer, C., Mignonneau, L. and Lopez-Gulliver, R. 2002. The Living Web - Immersive Web Environment. Presented at the Art-of-Immersion Festival at the ANIMAX Theather in Bonn, Germany. June 1 2002. http://www.art-of-immersion.com

Sommerer, C. and Mignonneau, L. 2002a. Modeling the Emergence of Complexity: Complex Systems, the Origin of Life and Interactive On-line Art, *LEONARDO Journal* (Cambridge/MA: MIT Press, Volume 35, Issue 2, pp. 161-169.

Sommerer, C. and Mignonneau, L. 2002b. Modeling Complexity for Interactive Art Works on the Internet," In: Art and Complexity: At the Interface, edited by J. Casti and A. Karlqvist, (New York: Perseus Books), in print.

Spector, L., and Alpern. A. 1994. Criticism, Culture, and the Automatic Generation of Artworks. In Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI-94, 3-8. Menlo Park, CA and Cambridge, MA: AAAI Press/The MIT Press.

Spiegelman, S. 1967. An in Vitro Analysis of a Replicating Molecule. *American Scientist*, Vol. 55, pp. 3-68.

Spofford, J. 1998. Primordial Life, available at: http://www.io.com/~spofford/index.html

Stengers, I. and Prigogine, I. 1983. Order Out of Chaos, New York: Bantam Books.

Stone, M. A. 1989. Chaos, Prediction, and Laplacean Determinism. *American Philosophical Quarterly* 26, pp. 123-131.

Taylor, C. E., D. R. Jefferson, S. R. Turner, and S. R. Goldman 1989, RAM: Artificial Life for the Exploration of Complex Biological Systems, In Artificial Life, edited by C. Langton. Santa Fe Institute Studies in the Sciences of Complexity Proc. Vol. VI. Addison-Wesley, Redwood City, CA.

Telefonica Spain. 1999, 2001. Vida/Life 2.0 and 3.0- Artificial Life Art competition: http://www.telefonica.es/fat/efat.html

Terzopoulos, D., Xiaoyuan T. and Grzeszczuk, R. 1995. Artificial Fishes: Autonomous Locomotion, Perception, Behavior, and Learning in a Simulated Physical World, In Artificial Life I, MIT Press, Vol. 1, No 4, pp. 327-351.

The Zen Room, 1998. A Global Multimedia and Internet Creative Cooperative - The Zen Room Limited, http://www.zenroom.demon.co.uk/

Thompson, A. 1998. Hardware Evolution: Automatic Design of Electronic Circuits in Reconfigurable Hardware by Artificial Evolution, Distinguished Dissertation Series, Springer Verlag.

Thompson, D'A. W. 1942. On Growth and Form. Cambridge: University Press.

Todd, P. M., and G. F. Miller 1991. Exploring Adaptive Agency II: Simulating the Evolution of Associative Learning, In From Animals to Animats, Proceedings of the First International Conference on Simulation of Adaptive Behavior, edited by J.-A. Meyer and S. Wilson. A Bradford Book, MIT Press, Cambridge and London.

Todd, S. and Latham. W. 1992. Evolutionary Art and Computers. Academic Press, London.

Trudeau, R. J. 1994. Introduction to Graph Theory. New York: Dover.

Turing, A. M. 1936. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings London Mathematical Society*. 42. pp. 230-265.

DeVany, A. 1993. Information, Bounded Rationality, and The Complexity of Economic Organisation. Discussion paper, Dept. of Mathematics, University of California.

Ventrella, J. 1995. Disney Meets Darwin-The Evolution of Funny Animated Figures, in Proceedings of Computer Animation 1995. Thalmann & Thalmann (eds), IEEE, pp. 35-43.

Ventrella, J. 1996. Sexual Swimmers: Emergent Morphology and Locomotion Without a Fitness Function, In From Animals to Animats, MIT Press, pp. 484.

Virtual Fishtank by NearLife, 1998. http://www.nearlife.com/projects/fishtank/fishtank.html

Von Neumann, J. 1956. The general and logical theory of automata, in J. R. Newman (ed.) The World of Mathematics, vol. 4, New York: Simon and Schuster, pp. 2070-2098.

Von Neuman, J. 1963. The General and Logical Theory of Automata, John von Neumann Collected Works, Volume V, A. H. Taub (ed.), Macmillan, New York, 1963, pp. 288-328.

Von Neuman, J. 1966. Theory of Self-Reproducing Automata. ed. and completed by A.W. Burks, Urbana/Champaign: University of Illinois.

Wächterhäuser, G. 1997. The Origin of Life and its Methodological Challenge. In: *J. Theoretical Biology,* 187, pp.483-494.

Waddington, C. H. 1966. Principles of Development and Differentiation. New York: Macmillan.

Waldrop, M.M. 1992. Complexity: The Emerging Science at the Edge of Chaos. New York: Simon and Schuster.

Waters, C. K. 1990. Why the Anti-reductionist Consensus Won't Survive the Case of Classical Mendelian Genetics. In Proceedings of the PSA, Vol. 1, East Lansing, pp. 125-139.

Weaver, W. and Shannon, C. E. 1949. The Mathematical Theory of Communication, Urbana, Illinois: University of Illinois Press (republished in paperback 1963)

Weisstein, E. 2002. Eric Weisstein's World of mathematics – A Wolfram Web Resource. http://mathworld.wolfram.com/

Whitelaw, M. 2000. Breeding Aesthetic Objects: Art and Artificial Evolution, in: Proceedings of the AISB'99 Symposium on Creative Evolutionary Systems (Edinburgh: Society for the Study of Artificial Intelligence and the Simulation of Behaviour), pp. 1-7.

Wicken, J. S. 1987 Evolution, Thermodynamics, and Information. Oxford University Press, Oxford.

Wimsatt, W. 1972. Complexity and Organisation. In K. Schaffner & R. Cohen (eds.) Studies in the Philosophy of Sciences. Dordretch: Reidel, pp. 67-86.

Winograd, S. 1963. Redundancy and Complexity of Logical Elements. *Information and Control*, Vol. 5, pp. 177-194.

Winslow, N. 1997. Introduction to Self-Organized Criticality & Earthquakes. Lecture Notes of the Dept. of Geological Sciences, University of Michigan, available at: http://www.geo.lsa.umich.edu/~ruff/Geo105.W97/SOC/SOCeq.html

Wisniewski, M. et al, 1999. netomat(TM). http://www.netomat.net/

Witten, I. H. and Frank E., 1999. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, The Morgan Kaufmann Series in Data Management Systems.

Woese, C.R. 1967. The Genetic Code - The Molecular Basis for Genetic Expression. New York: Harper and Row.

Wolfram, S. 1984. Cellular automata as models of complexity, *Nature* 311. pp. 419-424.

Wolfram, S. 1986. Theory and Applications of Cellular Automata. Singapore: World Scientific Press.

Wolfram, S. 1994. Cellular Automata and Complexity. Reading: Addison-Wesley.

Wyner, A.D. 1993. Collected Papers of Claude Elwood Shannon, N. J. A. Sloane and A. D. Wyner, editor, IEEE Press, New York.

Yaneer, B.-Y. 2000, New England Complex System Institute website: http://necsi.org/guide/whatis.html

Yaeger, L. 1994. Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, Vision, and Behavior or PolyWorld: Life in a New Context, in: Proceedings of the Artificial Life III Conference (held in 1992), ed. Chris Langton, Addison-Wesley , pp. 263-298.

Yates, F.E. 1978. Complexity and the limits to knowledge. *American Journal of Physiology*, Vol. 235, R201-R204.

Yates, F.E., ed. 1987. Self-Organizing Systems. The Emergence of Order. Plenum Press, New York.

Zeigler, B.P. 1976. Simulation based structural complexity of models. *International Journal of General Systems*, Vol. 2, pp. 217-223.

Zentner M. R. and Kagan. J. 1996. Perception of music by infants, *Nature*, 383, p.29.

Zurek, W.H. 1990. Algorithmic Information Content, Church-Turing Thesis, Physical Entropy, and Maxwell's Demon. In Zurek, W.H.(Ed). Complexity, Entropy and the Physics of Information. Redwood City, California: Addison-Wesley, pp. 73-89.

Zuse, H. 1991. Software Complexity. Berlin: Walter de Gruyter.

Zuse, H. and Bollmann, P. 1989. Using Measurement Theory to Describe the Properties and Scales of Static Software Complexity Metrics. *SIGPLAN Notices*, Vol. 24, No. 8, pp. 23-33.

Appendix

A) Life Spacies II - CD-ROM

A CD-ROM with the executable file of the "Life Spacies II" program was submitted together with this thesis. The software can run on any Macintosh system, optimal performance is obtained with a Mac G4 system.

Operational Instructions

1. Insert the CD-ROM in your computer.
2. Double click the Life Spacies icon.
3. A window with abstract plant picture will appear.
4. Now place the mouse cursor into the white area below this picture and write any text. Then press the return key.
5. A 3D creature will appear in the main window. Its genetic code is based on your text input.
6. To keep your creatures alive you need to feed it.
7. Give some food, by placing the cursor into the main window and then press any key on the keyboard. This will release the text characters you just typed. They are the food for your creature. Creatures will only eat text characters that are part of their genetic code. For example the creatures "John" will only eat "J", "o", "h" and "n" characters.
8. Food provides energy for your creatures and as long as you feed it, it can stay alive. Movement costs energy and if the creature moves a lot it has to eat frequently.
9. If a creature moves too much or does not eat enough it will sink to the floor and die.
10. If your creature has enough energy it can also mate with another creature. They will exchange their genetic code and a child creature can be born.
11. The child will look similar to its parents.

12. If you interact with the system well you can create a whole population of creatures, with new creatures being born and old creatures being discarded.
13. Due to systems resources management the number of creatures that can live at the same time has been limited to 20 for this desktop version of "Life Spacies II". However as some creatures may die, new ones can be created or born.

Please refer to Chapter 8 for detailed information on the "Life Spacies II" system.

Credits

Life Spacies II

Contact: christa@iamas.ac.jp, http://www.mis.atr.co.jp/~christa

B) Documentation CD-ROM

A CD-ROM with documenation of the various artworks as described in Chapters 8 and 9 was submitted. This CD-Rom contains:

1) Images of Life Spacies
2) Images of Life Spacies II
3) Movie Files of of Life Spacies
4) Movie Files of of Life Spacies II
5) Images of VERBARIUM
6) Images of Riding the Net
7) Press material about Riding the Net
8) Movie Files of Riding the Net
9) Images of The Living Room
10) Movie Files of The Living Room
11) Images of The Living Web
12) Move Files of The Living Web